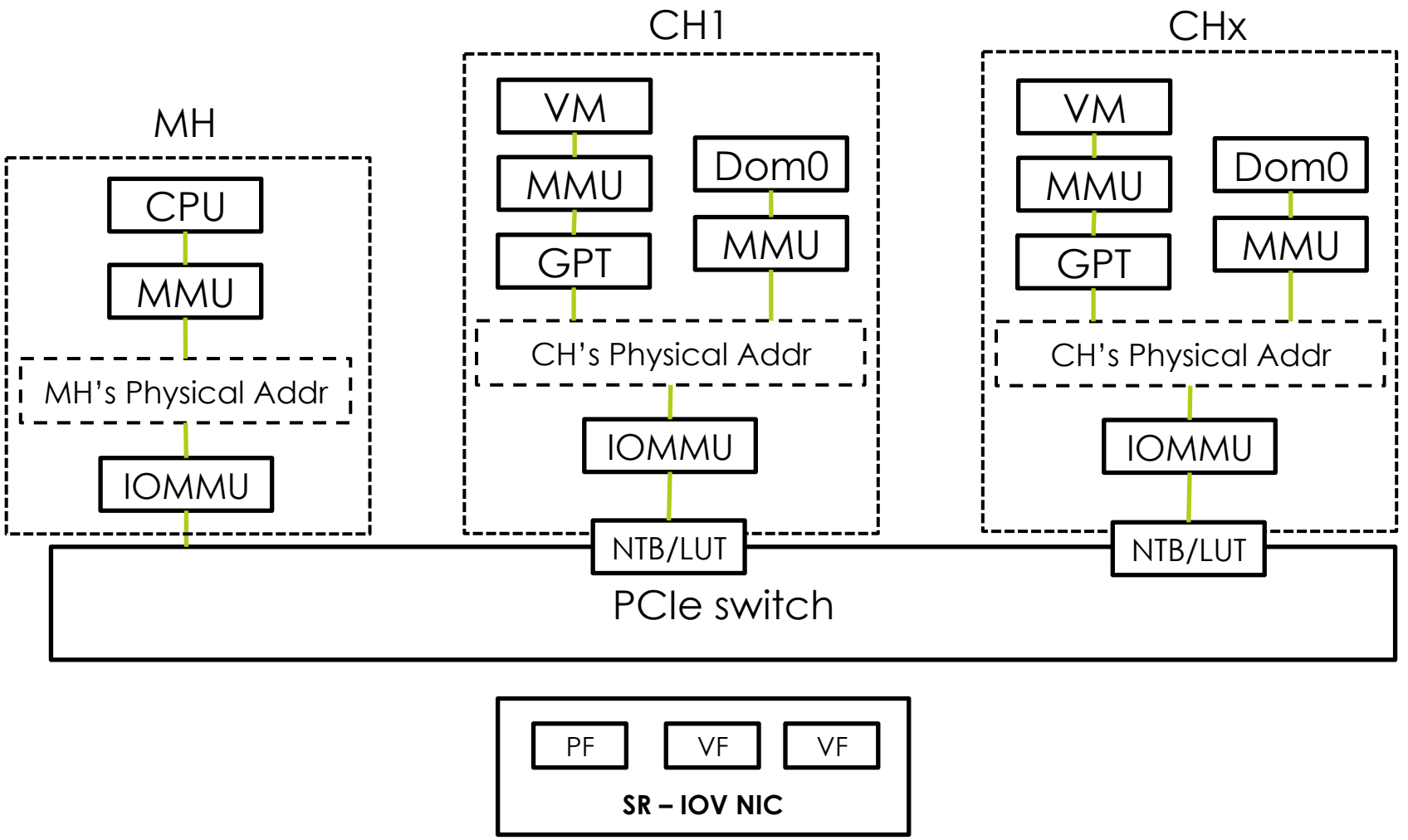
Cheng-Chun Tu

Secure Address Space Protection in a Multi-hosts Environment

V1. Sep 22, 2012

V2/V3. Sep 24, 2012

Architecture Overview



Outline

- What kinds of hosts and devices in the PCIe network?
- What kinds of address spaces in the PCIe network?
- What components do we use to translate among addresses space?
- How does the each address space being translated?

Hosts and Devices

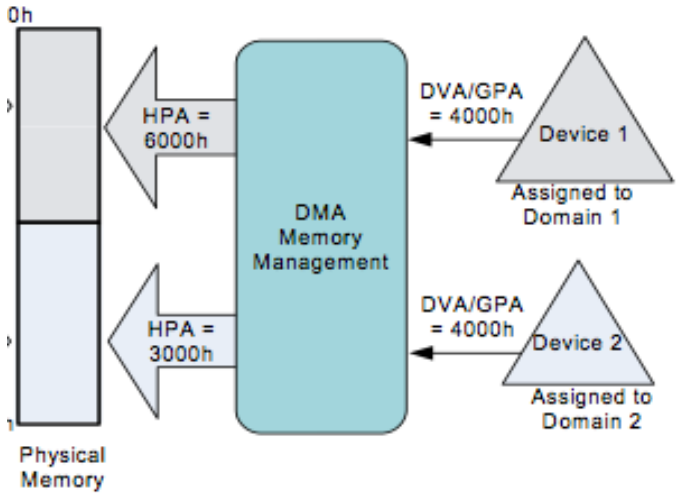
- MH (Management Host)
  - The host that extends its own PCI hierarchy into the PCIe switch fabric.
- CH (Computer Host): host behind a NT virtual side.
  - VCH (Virtualized CH): host with SW/HW supports for running virtual machines.
  - NVCH (Non-virtualized CH): host which runs single OS with IOMMU
- Devices
  - Normal PCI device
  - SR-IOV

Address Spaces in Hosts

- MH and NVCH
  - Physical address space
  - Virtual address space
  - Device virtual address space
- VCH
  - Host physical address space
  - Host virtual address space
  - Guest physical address space
  - Guest virtual address space
  - Device virtual address space

Address Space in Devices

- IOMMU provide each device its own device virtual address space
- IOMMU translates a device's virtual address into physical address by looking up the device's corresponding page table



- Host relies on IOMMU to isolation physical address range a device can access so that malicious device won't access privileged memory

Translation Components

Comp.	Identifier	input	Output
PT	Process ID	Host virtual addr.	Host phys addr
GPT	Process ID	Guest virtual addr.	Guest phys addr
EPT	Domain(VM) ID	Guest phys addr.	Host phys addr
IOMMU	Device ID	Dev virtual addr.	Host phys addr
LUT	Part of Device ID	Device ID	Device ID
NTB	BAR addresses	Physical Address at one side	Dev virtual addr. at the other side

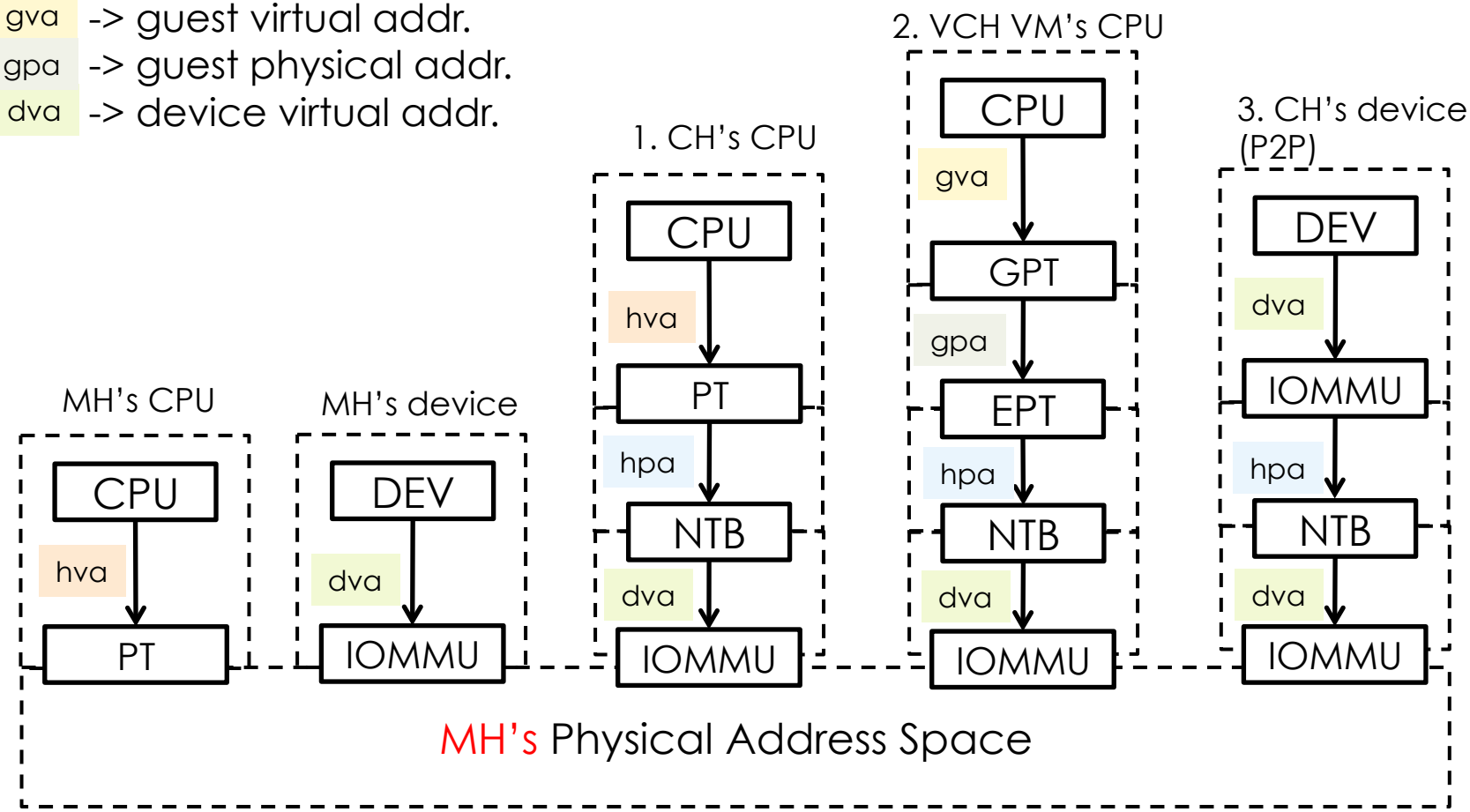
PT: Page Table
GPT: Guest Page Table
EPT: Extended Page Table
LUT: Look Up Table
NTB: Non-Transparent Bridge

NTB Address Mapping

- NTB maps from <the primary side to the secondary side>
- Mapping: <addrA at prim. side to addrB at the secondary>
  - addrA = Dev1 BARn, addrB = {RAM | Dev2 BARn}
  - Dev1: {NT-Link | NT-Virtual}
  - Dev2: {any device on the other side}
  - Actually addrB could target any secondary side physical address
- One-way Translation:
  - Read/write at addrA == read/write addrB
  - Read/write at addrB does not translated to addrA
- Type
  - addrA = {physical addr.}
  - addrB = {device virtual addr.}

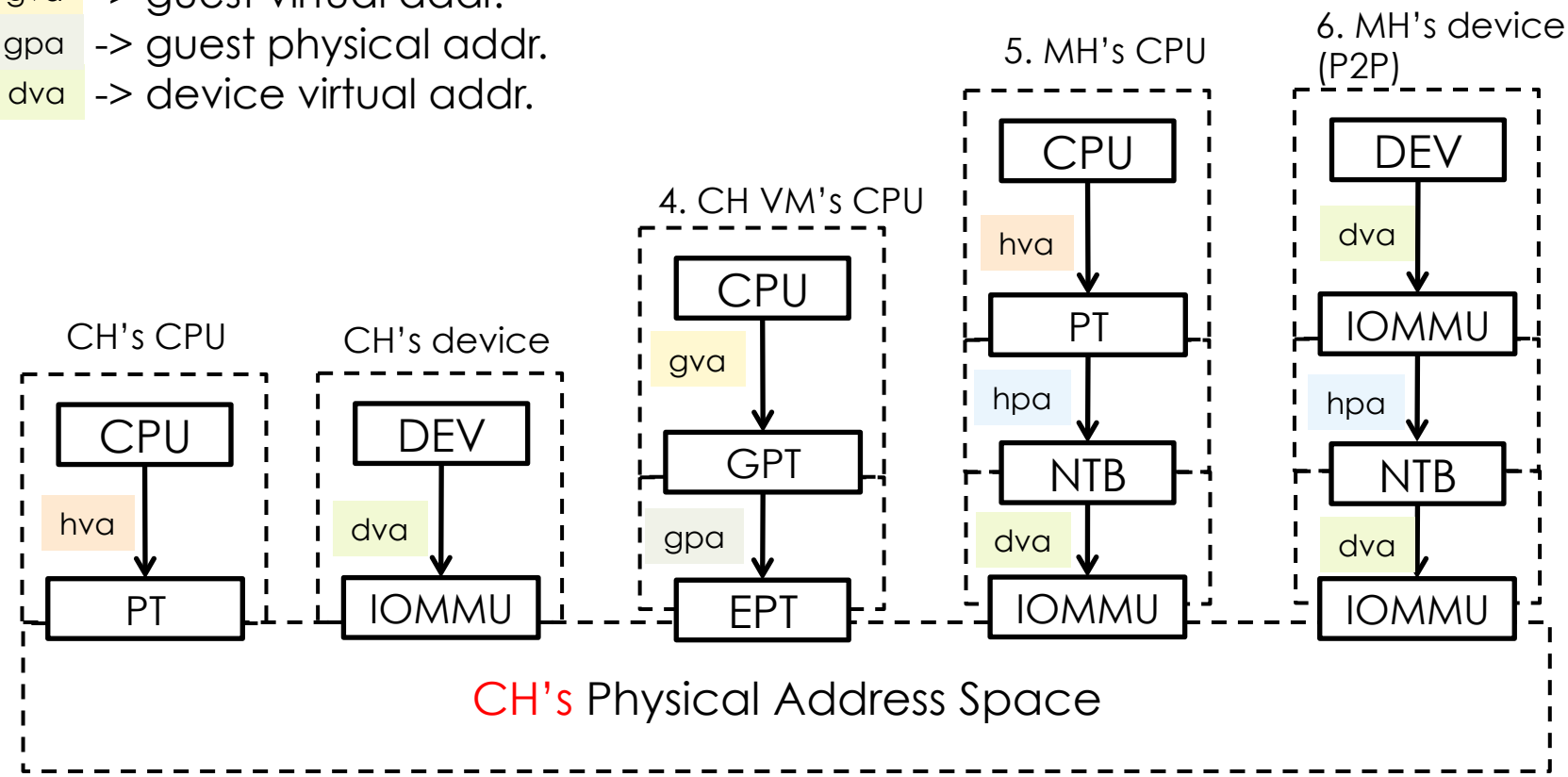
Address Translation to MH

hpa -> host physical addr.
hva -> host virtual addr.
gva -> guest virtual addr.
gpa -> guest physical addr.
dva -> device virtual addr.



Address Translation to CH

hpa -> host physical addr.
hva -> host virtual addr.
gva -> guest virtual addr.
gpa -> guest physical addr.
dva -> device virtual addr.



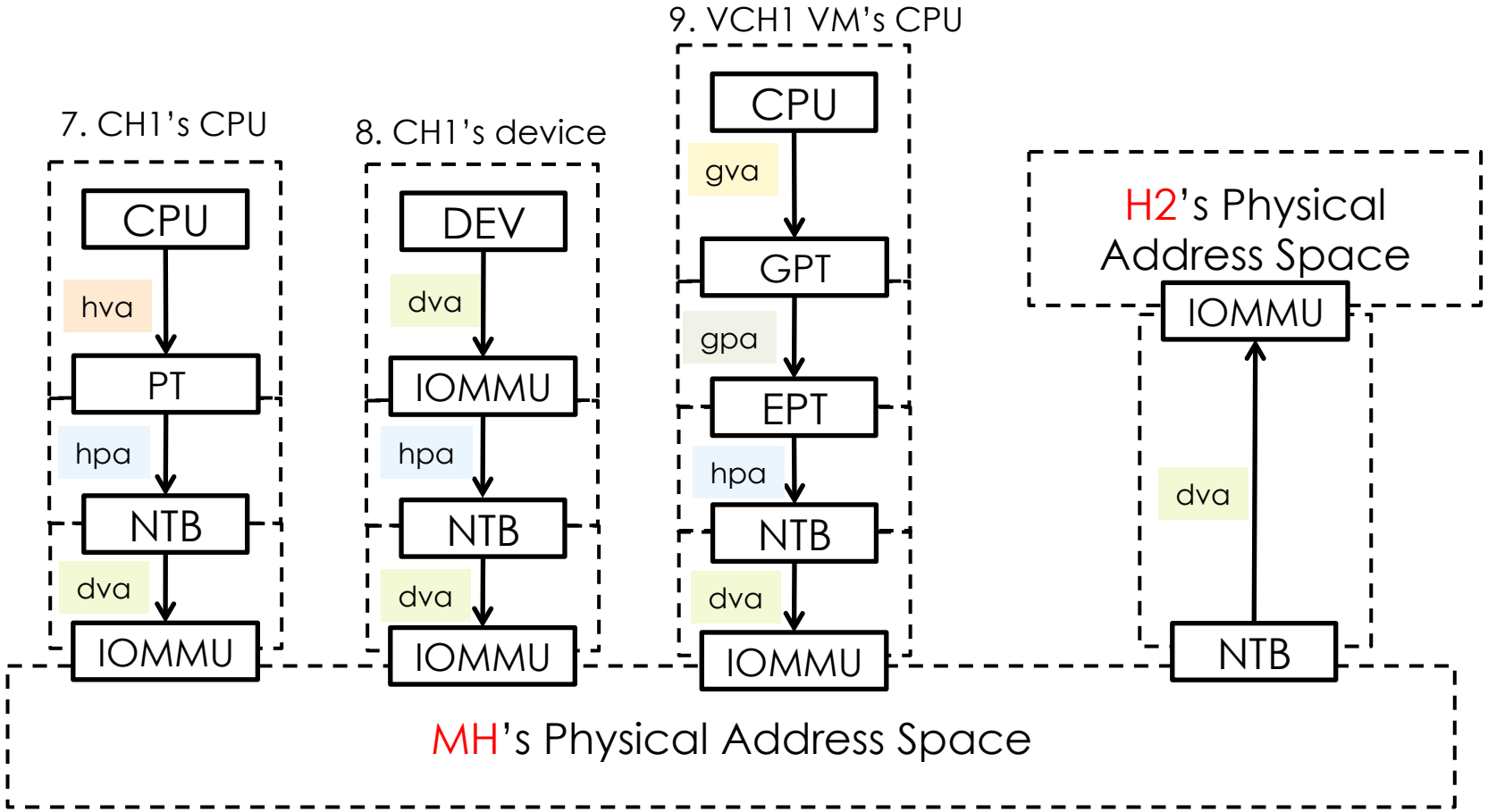
Summary 1

src \ dst	MH memory / device	CH memory / device
MH CPU	MMU	MMU+NTB+IOMMU
MH Dev	IOMMU	IOMMU+NTB+IOMMU
CH CPU	MMU+NTB+IOMMU	MMU
CH Dev	IOMMU+NTB+IOMMU	IOMMU
CH VM	MMU+GPT+NTB +IOMMU	MMU+GPT

Device-to-device protection requires switch ACS and P2P support.

Inter-host Address Translation

7. From CH1's CPU to CH2's RAM
8. From CH1's Device to CH2's RAM
9. From VM in CH1 to CH2's RAM



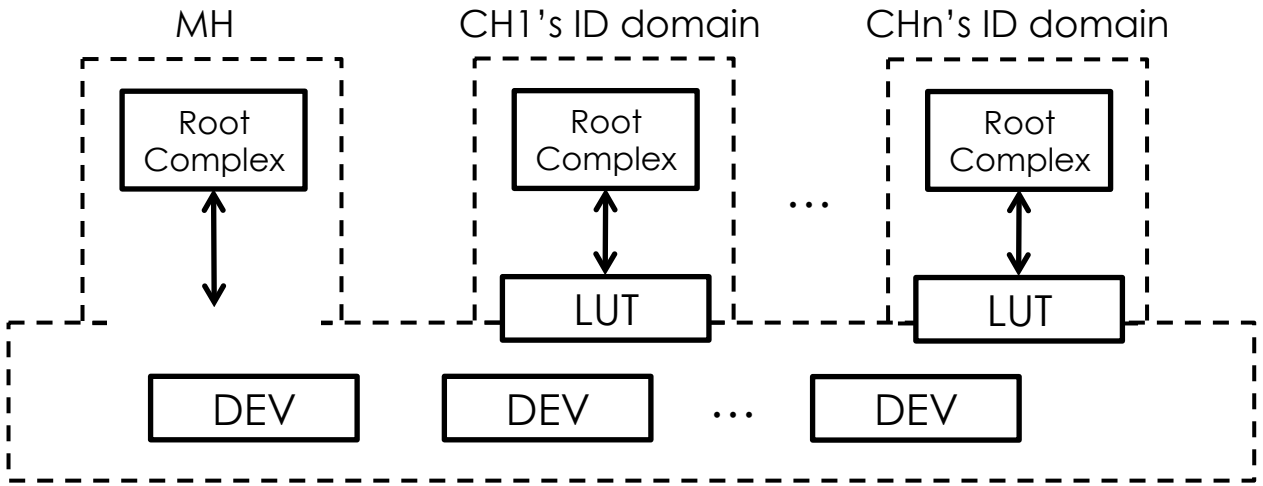
Summary 2

src \ dst	CH Memory	Other CH memory
CH CPU	MMU	MMU + NTB + MH's IOMMU + NTB + dst's IOMMU
CH Dev	IOMMU	IOMMU + NTB + MH's IOMMU + NTB + dst's IOMMU
CH VM	MMU + GPT	MMU + GPT + NTB + MH's IOMMU + NTB + dst's IOMMU

Case Description

Case	Description
1	CH accesses MH's memory,
2	VM on CH accesses MH's memory,
3	CH's device accesses MH's memory
4	CH's VM access CH's memory
5	MH's CPU accesses CH's memory
6	MH's device accesses CH's memory
7	CH1's CPU accesses CH2's memory
8	CH1's device accesses CH2's memory

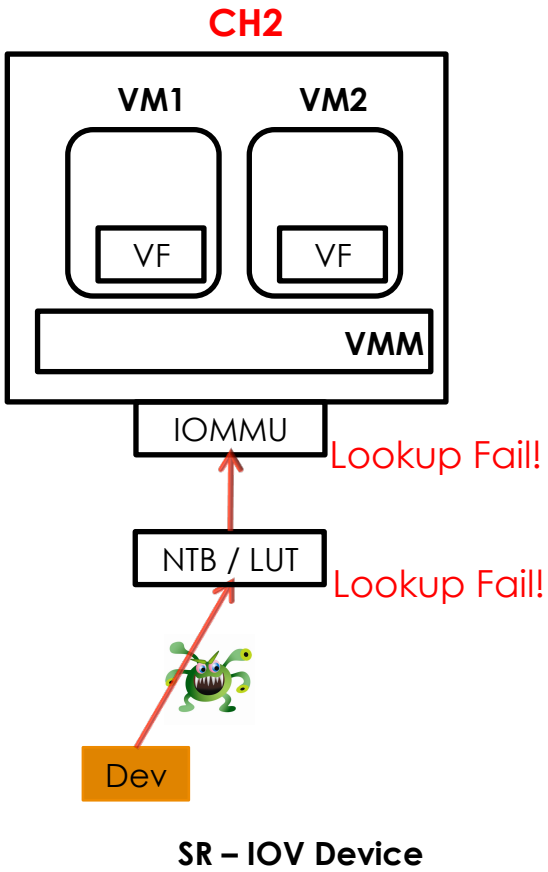
PCI device ID translation



1. Each host (MH + CHs) has its own device id domain.
2. LUT is responsible for ID translation from one host domain to another.
3. IOMMU depends on ID to setup its access table.

LUT + IOMMU based protection

- Prevent unauthorized dev from accessing any memory area in other CHs such as CH2
- Solution strategies:
  - LUT lookup fail or IOMMU lookup fail
- NTB translates dev's ID space from MH to CH2 by LUT.
- Make sure
  - Either LUT has no entry for dev's <bus:dev>
  - Or dev's translated ID is forbidden in the CH2's IOMMU table.



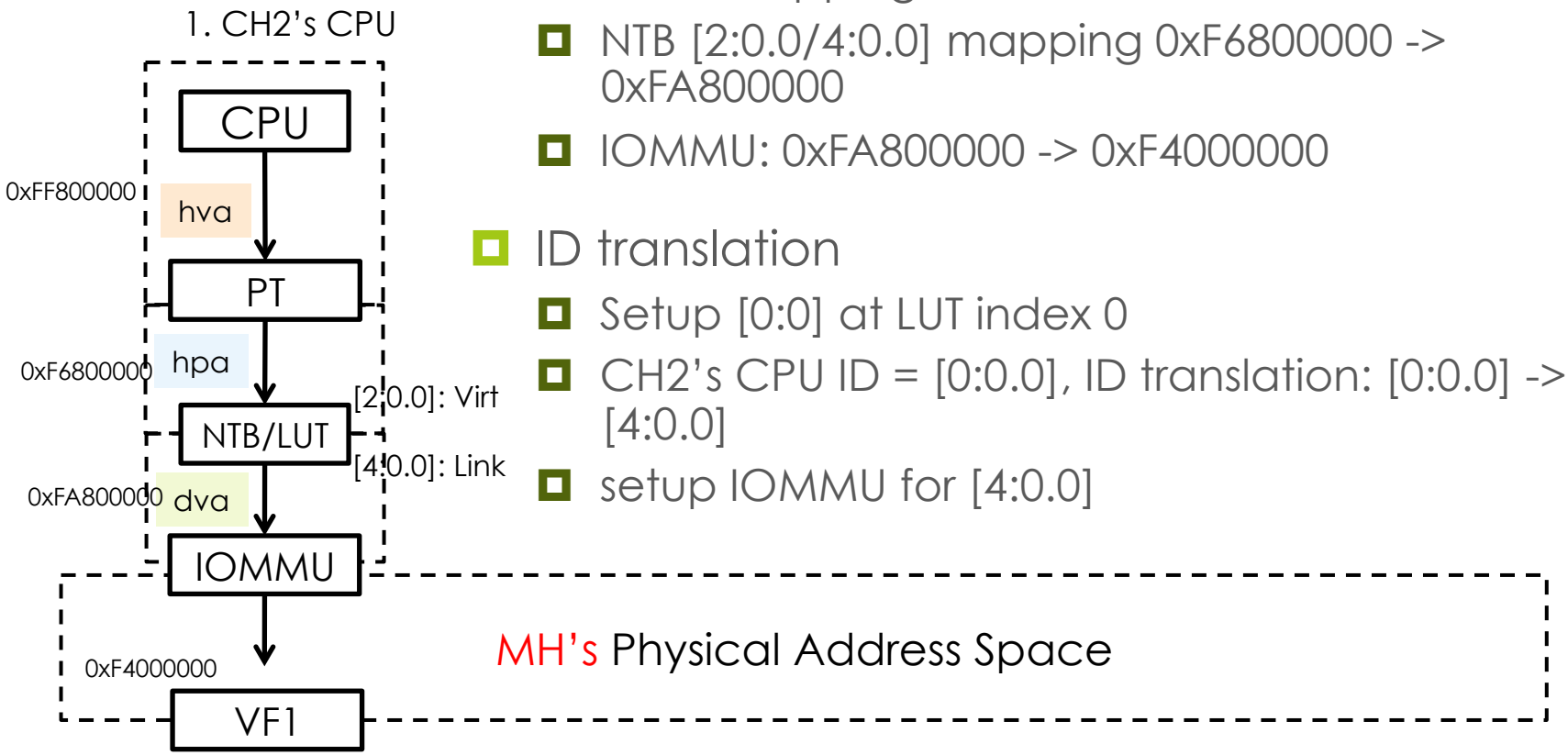
Examples

LUT translation example

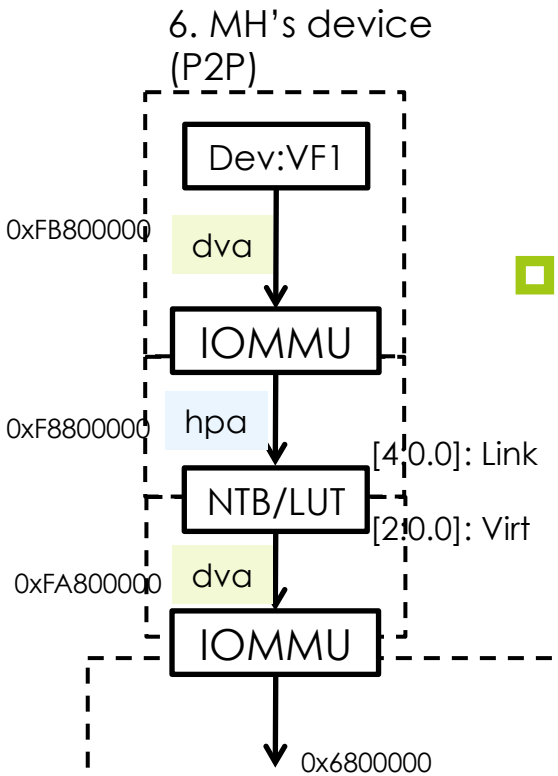
- Each VF / NTB is uniquely assigned a DevID <BusNo : DevNo . FunNo>
- LUT entries consists of **index : <BusNo, DevNo>**
  - LUT translation:
    - <BusNo : DevNo. FunNo> translate to
    - <NTB's BusNo : matching index in LUT. FunNo>
- Ex: VF1=<3:12.1>, VF5=<3:12.5>, NTB for VF5=<2:0.0>, LUT= 4 : <3:12>
  - VF1 after translation: <2:4.1>,  forbidden in IOMMU
  - VF5 after translation: <2:4.5>, allow in IOMMU

Example1 From CH2's CPU to MH's Device VF1's CSR

- MMU -> NTB -> IOMMU->MH VF1's CSR
  - VF1's CSR: 0xF4000000, ID=[3:11.1]
  - MMU mapping 0xFF800000 -> 0xF6800000
  - NTB [2:0.0/4:0.0] mapping 0xF6800000 -> 0xFA800000
  - IOMMU: 0xFA800000 -> 0xF4000000
- ID translation
  - Setup [0:0] at LUT index 0
  - CH2's CPU ID = [0:0.0], ID translation: [0:0.0] -> [4:0.0]
  - setup IOMMU for [4:0.0]



Example2 From VF1 to DMA CH2's memory 0x6800000

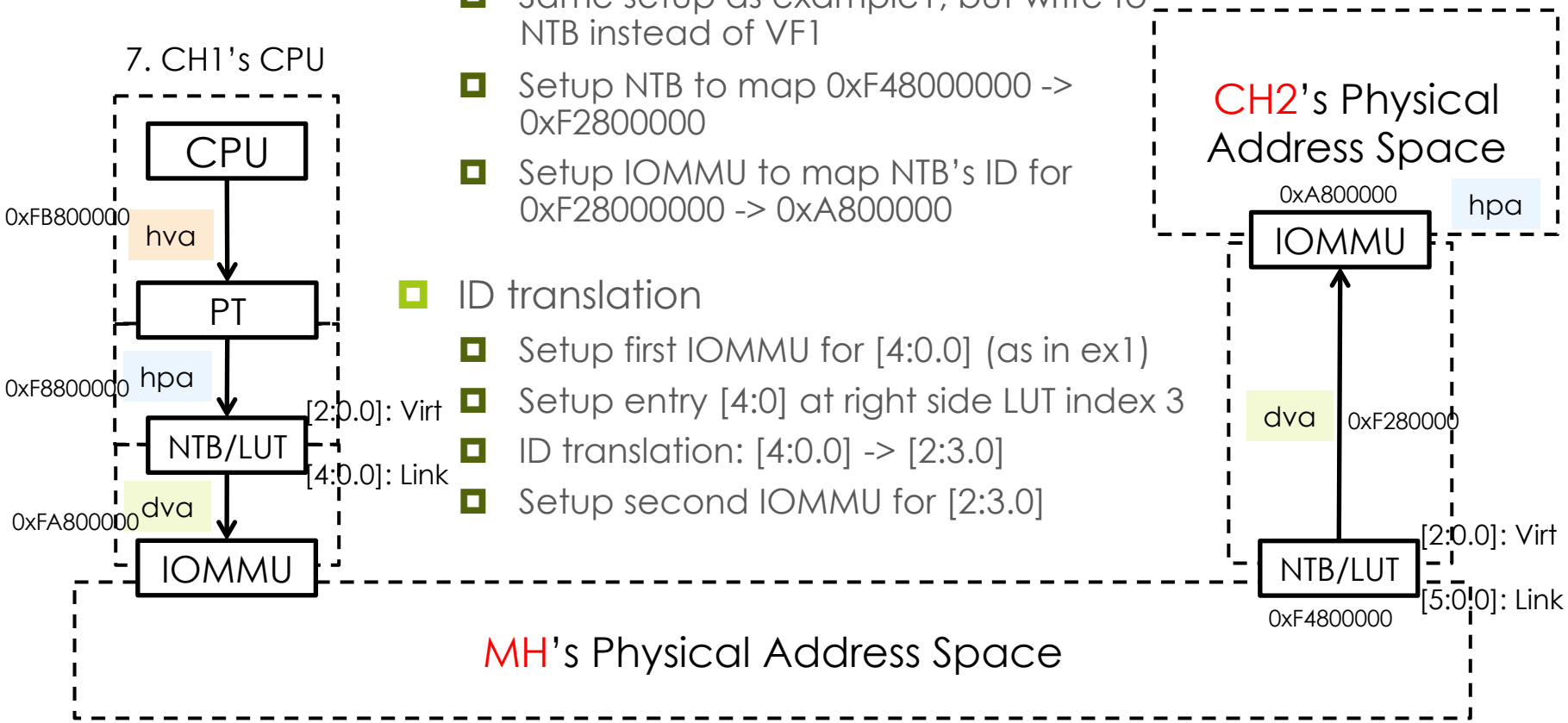


- VF1 -> IOMMU -> NTB -> IOMMU
  - VF1[3:11.2] writes 0xFB800000
  - IOMMU: 0xFB800000 -> 0xF8800000
  - NTB[2:0.0] maps 0xF8800000 -> 0xFA800000
  - IOMMU mapping 0xFA800000 -> 0x6800000
- ID translation
  - Setup first IOMMU for [3:11.2]
  - Setup entry [3:11] at LUT index 3
  - ID translation: [3:11.2] -> [2:3.2]
  - VF1's ID at CH2 = [2:3.2], setup second IOMMU for [2:3.2]

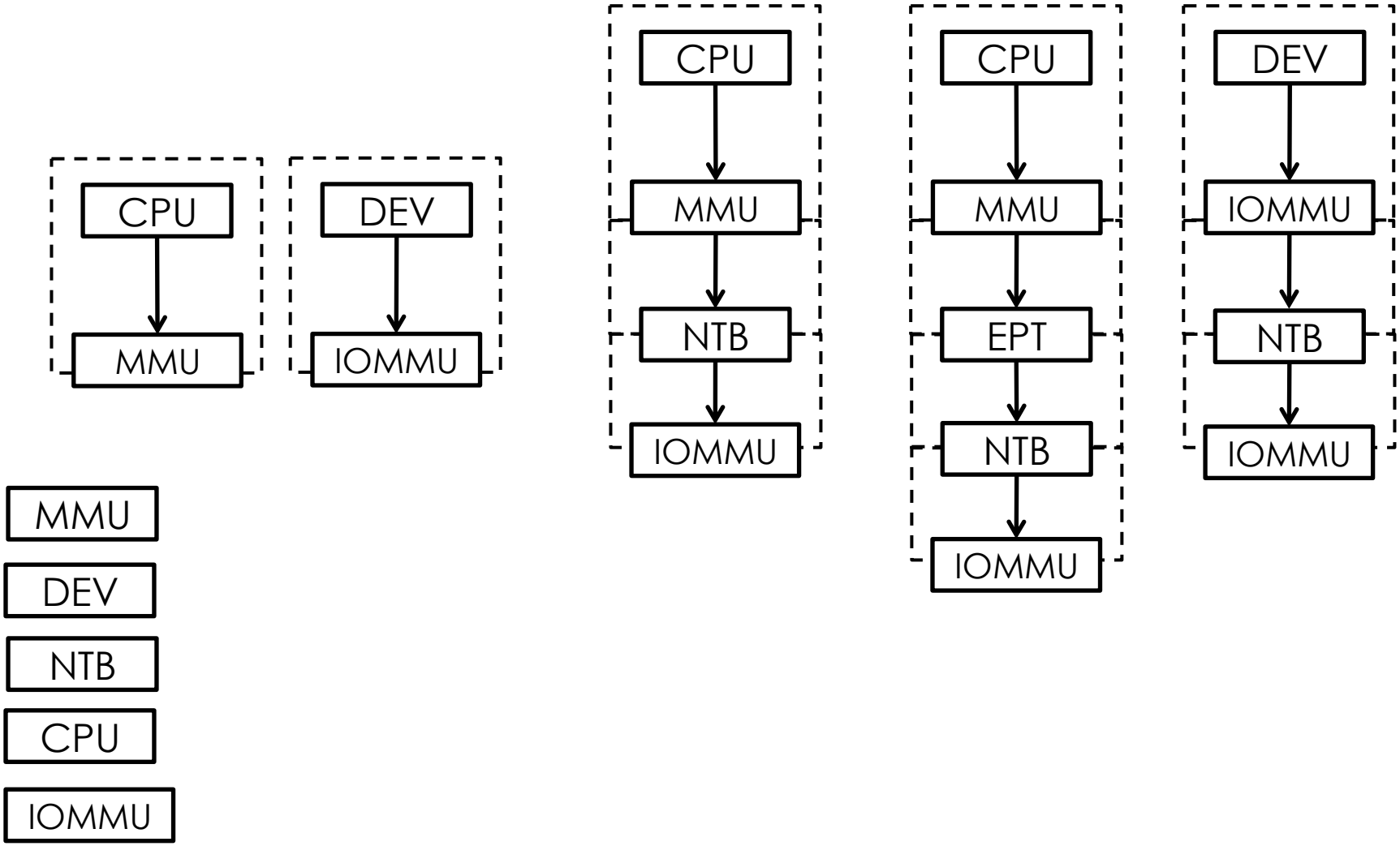
CH's Physical Address Space

Example3 From CH1's CPU to CH2's memory 0xA800000

- Let CH1 write to CH2 NTB's address mapping range
  - Same setup as example1, but write to NTB instead of VF1
  - Setup NTB to map 0xF48000000 -> 0xF2800000
  - Setup IOMMU to map NTB's ID for 0xF28000000 -> 0xA800000
- ID translation
  - Setup first IOMMU for [4:0.0] (as in ex1)
  - Setup entry [4:0] at right side LUT index 3
  - ID translation: [4:0.0] -> [2:3.0]
  - Setup second IOMMU for [2:3.0]



End


CPU
MMU
DEV
IOMMU
CPU
MMU
NTB
IOMMU
CPU
MMU
EPT
NTB
IOMMU
DEV
IOMMU
NTB
IOMMU
MMU
DEV
NTB
CPU
IOMMU

Translation Components

- PT (Page Table):
  - Host virtual address -> host physical address
- GPT (Guest Page Table):
  - Guest physical address -> Host physical address
- EPT (Extended Page Table):
- IOMMU:
  - Device virtual address -> Host physical address
- NTB:
  - Physical Address at one side of a host -> device virtual Address another side of a host
- LUT:
  - PCI device ID at one side of NTB -> PCI device ID at another side