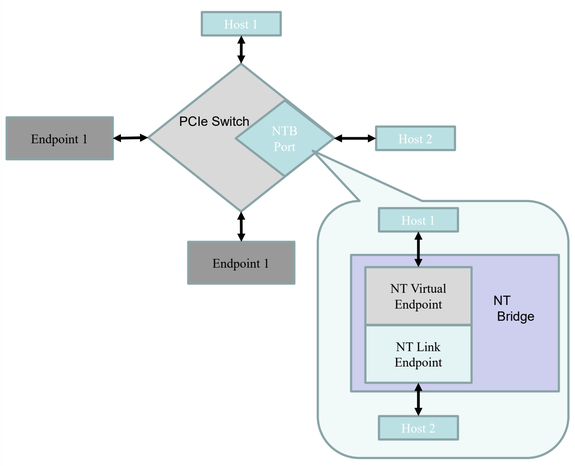
Improving Perfomance of the MTCA System by use of PCI Express Non-Transparent Bridging and Point-To-Point PCI Express Transactions

L.Petrosyan DESY MCS4

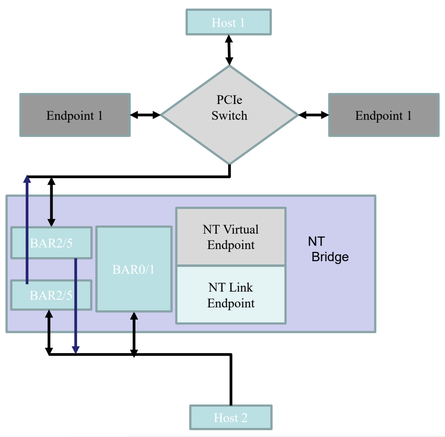
PCI Express NTB

- PCIe has to have only one Root Complex
  - PCIe Bus configuration and memory mapping
- NTB used to connect two independent address/Host domains
- A Non-Transparent bridge consist of two back-to-back PCIe endpoints, a Virtual and Link side endpoints.
- NTB isolates Address spaces of different Hosts by appearing as an endpoint to each side



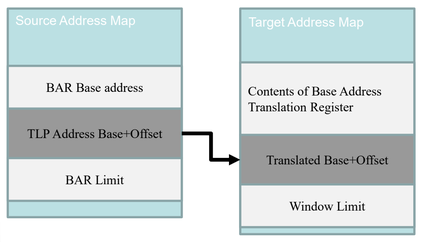
Key Elements of the NTB

- *BAR0/1 For configuration,* visible from both sides of the NTB
- *8 Scratchpad Registers (BAR0)*
  - provide a means of communication between two processors over a non-transparent bridge. They are readable and writeable from both sides of NTB.
- *16 Doorbell Registers (BAR0)*
  - The doorbell registers are used to send interrupts from one side of the NTB to the other.
- *Up to 4 BARS,* Individually disabled
  - *BAR2/5 are* apertures into the address space on the far side of the other endpoint, provides address transaction from one side to other

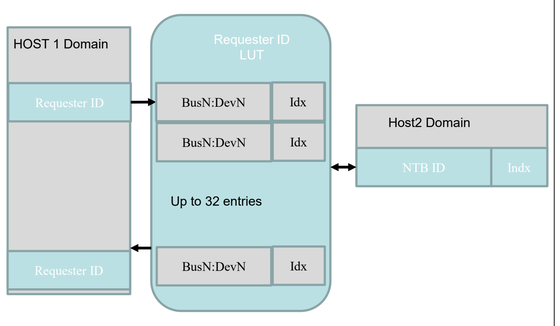


NTB Transactions

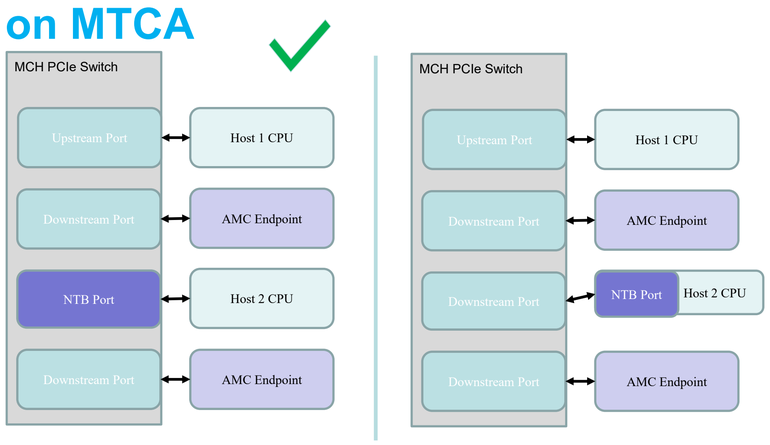
- **Direct Address Translation** for each enabled BAR on both sides
  - The content of the register could PCIe address of some endpoint in other side of the NTB



- **Requester ID conversion** (Bus Number, Device Number and Function Number) across NTB, Requester ID translation lookup tables



Different ways to enable NTB on MTCA



Setting up the NTB

Boot time configuration

1. Enable NTB on the Current port
2. Set up BARs
   - Enable BAR and set Size

Done by MCH
After boot the NTB ready for use on both sides.
From each side the predefined number of BARs are visible

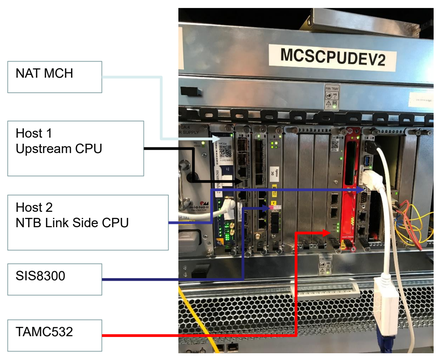
Run Time Configuration

1. Setup Address Translation Register for current BAR
2. Setup Requester ID Look Up Tables

NTB Virtual and Link side device drivers communicate using Scratchpad Register and Doorbell Register to share the information.
The drivers setup Address Translation Tables and Requester ID Look Up Tables.
The drivers initiate PCIe Transactions.

STATUS

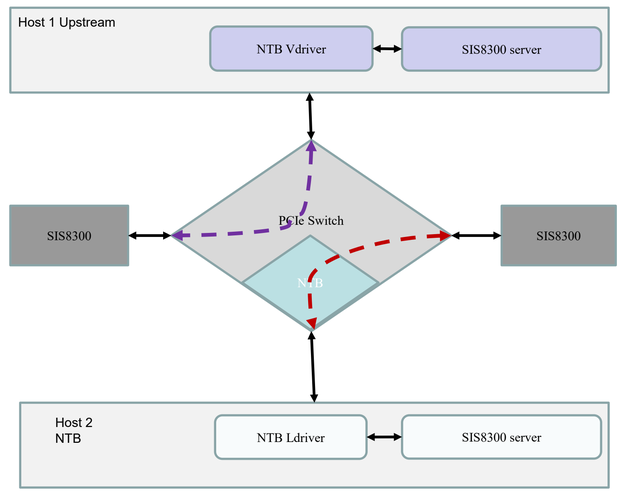
• PCIe Switch on MCH configured to have 2 BARS on each NTB side with 1MB size
• Used Virtual and Link side NTB device drivers created on base of Universal driver
• Universal driver has information about all AMC PCIe endpoints
• Virtual/Link side Driver sets Address Translation Registers:
  • Link Side BAR2 address of SIS8300
  • Link Side BAR3 address of TAMC532
• Virtual Side Driver sets RID-LUT:
  • Virtual Side LUT: Root Complex
  • Virtual Side LUT: SIS8300
  • Virtual Side LUT: TAMC532
• Link side Driver sets RID-LUT
  • Link Side LUT: Root Complex
	Boot Time-MCH
	
	Run Time NTB Device Drivers On both sides
	
	NTB Virtual and Link side device drivers communicate using Scratchpad Register and Doorbell Register to share the information.

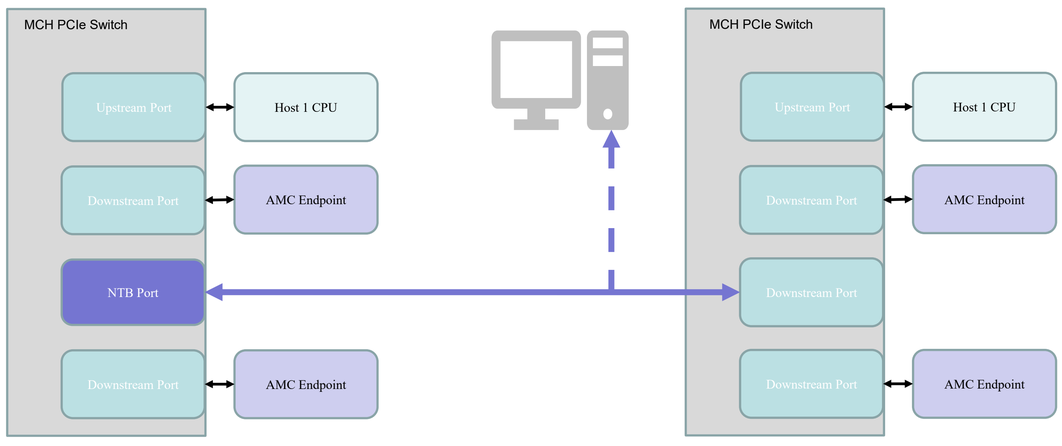


Transaction from Host2 to SIS8300 and TAMC532 are tested

Different Use Cases

- Link Side server controls one of AMC Module
- Virtual Side server controls both modules
  - One module send Data to Virtual Side
  - Other module sends Data to Link Side using PCIe Poin2 Point connections
  - Connect External CPU





- The source codes can be found on https://github.com/MicroTCA
- The information and Linux packages can be found on a DOOCS web page ***http://doocs.desy.de***
- Mail ***doocs@desy.de***