PCI PASID support in arm SMMUv3

- SID ("Stream ID") identifies a device (32 bits). Roughly equivalent to PCI RID with PCI domain information in the top bits.
- SSID ("Substream ID") = PASID (20 bits)
- Context descriptor tables = PASID tables.
- Depending on the configuration, PASID0 may be reserved and the first context descriptor used for non-PASID transactions.
PCI ATS and PRI support in arm SMMUv3

Interface stays very close to PCI:

- ATC invalidation
  (SID, PASID, address, size, Global flag)

- PRI Page Request
  (SID, PASID, PRG, address, Last/W/R/X/Priv flags)

- SMMU driver sends a PRI_RESP command
  (SID, PASID, PRG, success/invalid/failure)
Arm systems, especially embedded ones, have DMA masters on the system bus, not behind a PCI bridge.

PASID use is identical to PCI.

But they don’t do ATS/PRI. They use the ”stall” model.
Platform SVM - the stall model

- Memory read/write requests that fail SMMU translation can be stalled indefinitely.
- The fault is reported on the Event queue (SID, PASID, tag, address, translation fault information) Same as a PPR with a little more details.
- SMMU driver sends a RESUME command (SID, tag, retry/terminate) Same as a PRI Response (but PASID is implicit)
- No ATC invalidation, only SMMU TLB.
- Driver implementation reuses most of the PRI code.
- No API change needed.
Questions

The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks