Alternative Requester ID Interpretation (ARI)

Michael Krause (HP)

Disclaimer

- This presentation covers a proposed ECR being developed for the PCI Express Base Specification.
  - ✓ The ECR is applicable to both Base and IOV capable PCI Express components.
- The ECR is still under development and subject to change based on feedback from the PCI-SIG members and the attendees at this training event.
- Please provide feedback at any time within this presentation or training event.

Requester ID (RID) Review

- A Requester ID is a 16-bit field composed of three elements:
  - ✓ Bus Number – 8 bits in length
  - ✓ Device Number – 5 bits in length
  - ✓ Function Number – 3 bits in length
- Functions must capture the Bus and Device Numbers supplied with all Configuration Write Requests (Type 0) completed by the function and supply these numbers in the Bus and Device Number fields of the Requester ID for all Requests initiated by the device/function.
  - ✓ The Bus Number and Device Number may be changed at run time, and so it is necessary to re-capture this information with each and every Type 0 Configuration Write Request.
- Each function associated with a logical device must be designed to respond to a unique Function Number for Configuration Requests addressing that logical device.
  - ✓ Each logical device may contain up to eight logical functions.

PCIe Constraints

- PCIe Base specification v1.1 states:
  - ✓ All PCI Express components are restricted to implementing a single Device Number on their primary interface (Upstream Port), but may implement up to eight independent functions within that Device Number.
  - ✓ Switches and Root Complexes must associate only Device Number 0 with the Endpoint attached to the logical bus representing the Link from a Switch Downstream Port or a Root Port.
  - ✓ Configuration Requests specifying all other Device Numbers (1-31) must be terminated by the Switch Downstream Port or the Root Port with an Unsupported Request Completion Status (equivalent to Master Abort in PCI).

Problem Statement

- PCIe Base specification constraints impeding innovation
  - ✓ New process technology is enabling developers to integrate an increased number of I/O functions within a single Endpoint
    - PCI-SIG member companies prefer to avoid incorporating one or more virtual switches to provide increased fan-out as noted in the specification.
      - Do not want to incur virtual switch cost / complexity
      - Do not want to consume additional Bus Numbers
  - ✓ Virtualization requires each SI to see a unique I/O identifier
    - Each Virtual Endpoint must be assigned a unique RID
      - Multiple Endpoints may be assigned per SI
      - Multiple Virtual Endpoints may be assigned per SI
    - Given advent of multi-core processors, sub-CPU sharing, etc. the number of SI per RC will increase making the current PCIe constraints non-tenable in the long-term

Goals

- No change to the PCIe TLP wire protocol
  - ✓ Requester ID remains a 16-bit field
- Minimally, bound number of enumeration operations
  - ✓ Avoid issuing enumeration operations to potentially large RID space per Endpoint
- Optimally, define a deterministic set of Functions
  - ✓ Enumeration discovers all Functions through defined method
- Enable IHV to optimize resources if Functions are not used

High-level Approach

- A Requester ID is a 16-bit field composed of two elements:
  - ✓ Bus Number – 8 bits in length
  - ✓ Function Number – 8 bits in length
- Function 0 is required
  - ✓ This is the equivalent of Device Number = 0, Function Number = 0
    - – Required to enable variety of PCIe configuration operations to continue to operate as they do today
- An Endpoint can be assigned multiple Bus Numbers
  - ✓ Additional Bus Number bits are concatenated with the Function Number bits to identify a Function
    - – Sparse Function Number space is allowed
    - – Number of Functions per Bus Number is variable
      - Not required to be uniform across all Bus Numbers
  - ✓ Endpoint would be required to respond to Type 1 Configuration transactions on the additional Bus Numbers

ARI Requirements

- ARI is optional normative capability and control registers applicable to:
  - ✓ Switches
  - ✓ Root Ports
  - ✓ Endpoints
    - – Single Function, Multi-Function, Root Complex Integrated
- ARI must be backward compatible with existing hardware and software
  - ✓ If the capabilities are not enabled, the hardware and software operate in compliance with the existing PCIe 1.1 Base specification
- ARI capable software should be capable of:
  - ✓ Enabling / disabling ARI per component type
  - ✓ Determining the maximum number of Bus Numbers supported
  - ✓ Setting the Bus Number range configured within the component for use by ARI
  - ✓ Determining the maximum number of Function Numbers per Bus Number supported
  - ✓ Setting the maximum number of Function Numbers allowed per Bus Number
  - ✓ Assigning a Function Number to a Function Group if arbitration is allowed.

ARI Requirements (cont.)

- If ARI is configured, Phantom Function Numbers are not supported
  - ✓ The Phantom Function bits within an Endpoint's Device Register (see Base spec Section 7.8.3, Table 7-11) must be set to 00b
  - ✓ An Endpoint can still use the Extended Tag Field Enable bit (see Base spec Section 7.8.4, Table 7-12) to enable each Function to support up to 256 outstanding requests.
- If ARI is configured and the Endpoint or Root Complex Integrated Endpoint supports multi-function arbitration, the Functions must be configured into Function Groups.
  - ✓ A maximum of eight Function Groups may be configured per Endpoint or Root Complex Integrated Endpoint.
  - ✓ Arbitration is accomplished on a Function Group basis per arbitration phase using the same controls as multi-function arbitration.

Example ARI Enablement Flow

- Software enumerates the PCI Express hierarchy and determines whether the ARI capability is supported. ARI has no impact on the base enumeration algorithms used in platforms today.
  - ✓ For a Switch or a Root Port, the capability is communicated through fields being defined via the PCIe Capability Structure Expansion ECN
  - ✓ For an Endpoint or a Root Complex Integrated Endpoint, the capability is communicated through a new PCI Express extended capability.
    - This capability is accessed only through Function Number = 0.
- Software sets the ARI enablement bits in each component.
  - ✓ For a Switch or a Root Port, this involves a single bit.
    - Setting this bit alters the default semantics described in section Base specification section 7.3 to enable non-zero Device Numbers to be forwarded to an Endpoint without triggering an error
  - ✓ For an Endpoint or a Root Complex Integrated Endpoint software configures the appropriate Bus Number range and the maximum number of Function Numbers allowed depending upon the resource capabilities supported.

Options Being Explored

- Several options are being explored to realize these requirements:
  - ✓ Replicate capability and control registers per Function
  - ✓ Use MMIO space to contain all ARI structures
  - ✓ Define a bit mask per Bus Number that contains:
    - A bit per Function present
    - A bit per Function enabled
  - ✓ Use a configuration space structure that enables a “window” into Endpoint resources
    - Window would allow software to access a subset of resources
  - ✓ Etc.
- Any feedback on any approach is appreciated.

ARI Function Group Arbitration

- Goal is to largely re-use the existing multi-function arbitration mechanisms and structures with following minor changes
  - ✓ Each arbitration slot represents a Function Group
    - Existing RR, WRR, etc. schemes remain unchanged
    - On each slot period, a Request is “pulled” from the associated Function Group and transmitted on the link
  - ✓ Function Numbers are assigned to a Function Group
    - Method of assignment is TBD
      - Will support hardware defined mechanism as default
      - Will support software assigned mechanism
  - ✓ Arbitration within each Function Group itself is not defined
    - Hardware specific, e.g. FIFO

Summary

- ARI is applicable to PCIe Base and IOV-capable components
- ARI is an optional normative capability
  - ✓ Interoperable with existing PCIe 1.x Base components
- ARI re-uses the existing 16-bit RID field
  - ✓ There are no TLP wire protocol changes
- ARI removes constraints / re-interprets the RID to enable additional Functions to be identified within an Endpoint
  - ✓ This enables potentially thousands of Functions to be assigned to a given Endpoint
  - ✓ For IOV, ARI enables greater than 8 SI to share an Endpoint

Thank you for attending the
PCI-SIG IOV Training Event.

For more information please go to
www.pcisig.com