
PCI
SIG®

IOV Errors and Events

Eric DeHaemer (Intel)

Outline

- Error Reporting
- Interrupts
- Other Events

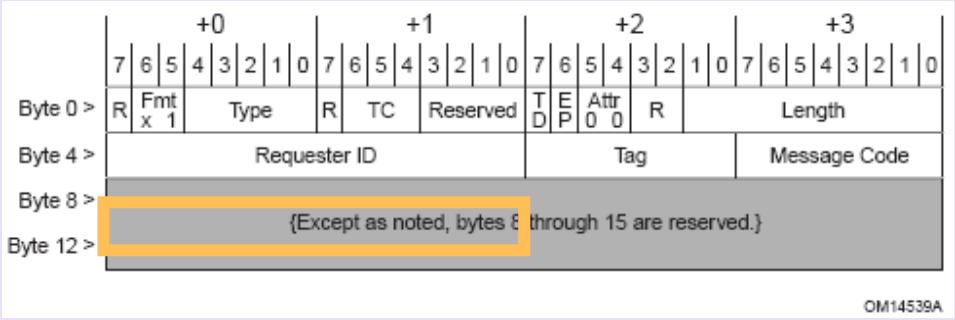
Base Specification Error Reporting

Background

- Two error reporting paradigms
  - ✓ the baseline capability –required by all PCIe devices
  - ✓ the Advanced Error Reporting capability - optional
- Baseline error reporting capabilities define the minimum error reporting requirements.
- The Advanced Error Reporting is defined for more robust error reporting

Error Messages

- Error Messages are sent to the Root
  - ✓ ERR_COR
  - ✓ ERR_NONFATAL
  - ✓ ERR_FATAL
- Initiator of the Message identified by the Requestor ID of the message header
  - ✓ Error Source Identification Register – with Advanced Error Reporting
- Errors Messages translated into platform level events
  - ✓ System Error
  - ✓ Error Interrupt – with Advanced Error Reporting



Baseline Error Reporting

- Required by all PCI Express devices
- Uses conventional PCI error status bits
  - ✓ Master Abort, Signaled Target Abort, Master Data Parity Error, etc.

Advanced Error Reporting

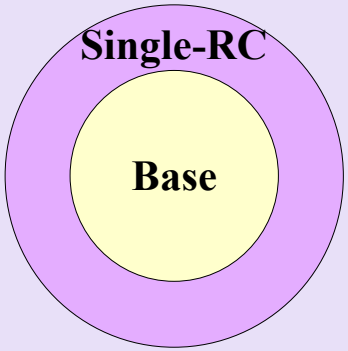
- Device Logs Header of TLP that caused the Error
  - ✓ Correctable / Uncorrectable status bits are set
  - ✓ Conventional status bits are also set (Master Abort, Target Abort, etc.)
- Root Port logs the Source ID found in the Error Message
  - ✓ Logged in the Error Source Identification register.
  - ✓ Records the Requester ID of the first ERR_NONFATAL/ERR_FATAL (uncorrectable errors) and ERR_COR (correctable errors)
- The Root Error Command register allows control of response to error Messages
  - ✓ Independent interrupt control for the three types of error Messages.

Multi-Function Device

- In a multi-function device, errors that are not related to any specific function, are logged in the status and logging registers of all functions.
- The following PCI Express errors are not function-specific:
  - ✓ All Physical Layer errors
  - ✓ All Data Link Layer errors
  - ✓ These Transaction Layer errors:
    - ECRC Fail
    - UR, when caused by no function claiming a TLP
    - Receiver Overflow
    - Flow Control Protocol Error
    - Malformed TLP
    - Unexpected Completion
- On the detection of one of these errors, a multi-function device should generate at most one error reporting Message of a given severity.
- Software is responsible for scanning all functions in a multi-function device when it detects one of these errors.

Single Root Error Virtualization

SR Error Logging



- Want to identify the affected SI
  - ✓ Which Virtual Function
- IOV devices are fundamentally Multi-Function Devices
  - ✓ Physical functions plus Virtual Functions
- Evolutionary approach over Base
- Errors are logged according to Base Specification rules

SR Baseline Error Control

- Baseline Error mechanism is required for PF and VF
- VI owns RC error registers and is first responder on errors
- Baseline Error Control bits are not replicated per VF
  - ✓ Per device control
  - ✓ Reads of VF space return the value of the PF
    - SERR# Enable
    - Parity Error Response
    - Correctable Reporting Enable
    - Non-Fatal Reporting Enable
    - Fatal Reporting Enable
    - UR Reporting Enable

SR Baseline Error Status

- Baseline Error Status bits are replicated per VF
  - ✓ Help with fault isolation
  - ✓ Per VF error detection and logging
    - Master Data Parity Error
    - Signaled Target Abort
    - Received Target Abort
    - Master Abort
    - SERR# Asserted
    - Detected Parity Error
    - Correctable Error Detected
    - Non-Fatal Error Detected
    - Unsupported Request Detected

SR Advanced Error Control

- Advanced Error Reporting still optional
  - ✓ Improves fault detection/isolation
- Error Mask, and Severity bits are not replicated per VF
  - ✓ Per device control
  - ✓ Reads of VF space return the value of the PF
    - Uncorrectable Error Mask Register
    - Uncorrectable Error Severity Register
    - Correctable Error Mask Register
    - ECRC Generation Enable
    - ECRC Check Enable

SR Advanced Error Status

- Non-Function Specific Errors are logged in the PF
  - ✓ Error logged in one register only
  - ✓ VI avoids touching all VFs to clear device level errors
  - ✓ The following errors are not function specific
    - All Physical Layer errors
    - All Link Layer errors
    - ECRC Fail
    - UR, when caused by no function claiming a TLP
    - Receiver Overflow
    - Flow Control Protocol Error
    - Malformed TLP
    - Unexpected Completion

SR Advanced Error Status

- Subset of Status bits are replicated per VF
  - ✓ Per VF error detection and logging
  - ✓ Help with fault isolation
  - ✓ The following errors are function specific
    - Poisoned TLP received
    - Completion Timeout
    - Completer Abort
    - UR, when caused by a function that claims a TLP
    - ACS Violation

Open Issues

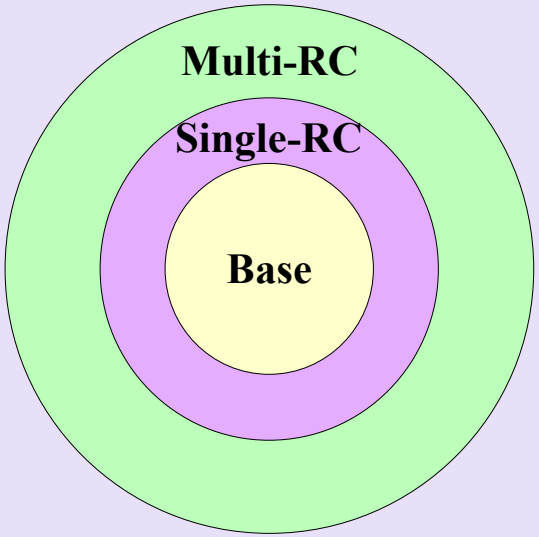
If Implemented, can VFs share the same Adv Error Header Log with the PF?

- ✓ Could Eliminate AER Header Log (128bits) per VF
- ✓ Reads of VF return log from PF
- ✓ Header log includes RID which identifies VF

- If AER is implemented in the PF is it still optional in the VF
  - ✓ Lower the bar for a device to implement AER
  - ✓ Useful for devices that have a specialized driver associated with the PF that is responsible for device initialization and error recovery

Multi-Root Error Virtualization

Multi-Root Error Logging



- Want to alert the affected Root Port(s)
  - ✓ Which Virtual Hierarchy
- Builds on Base and SR-PCIM.
  - ✓ Concentric circles
- MR devices are fundamentally Multi-Function Devices
  - ✓ Multiple Physical functions
  - ✓ May include Virtual Functions
- Leverage Multi-Function mechanisms

Multi-Root Error logging

- Each Virtual Hierarchy has a Physical Function
  - ✓ Physical functions have independent configuration space
  - ✓ Baseline Error control and status bits are present (required)
  - ✓ Advanced Error Reporting may be present (optional)
- Virtual Hierarchy of MR-PCIM is known by device

Base Errors

- Errors that are not related to any specific **virtual hierarchy:**
  - ✓ are logged in the status and logging registers of all active virtual hierarchies including MR-PCIM (VH0).
  - ✓ The following PCI Express errors are not VH specific:
    - – All Physical Layer errors
    - – All Data Link Layer errors
- Most Transaction layer errors are VH Specific
  - ✓ Contain a valid VH number
  - ✓ Error sent to affected Root Port
  - ✓ Base Multi-Function rules apply within the Virtual Hierarchy
- Some Transaction Layer errors are also relevant to MR-PCIM
  - ✓ Send to affected Root Port(s) and MR-PCIM
    - – Receiver overflow
    - – Flow Control Protocol Error

MR Specific Errors

- Some errors impact all hierarchies
  - ✓ Sent to all active VH including MR-PCIM
    - Invalid VH number
    - Error in training/flow control
    - Additional errors
  - ✓ Analogous to Multi-Function Error in base
- Some errors impact a single VH
  - ✓ Sent to affect VH and MR-PCIM
    - Error in training/flow control
- Some errors used to notify MR-PCIM of fabric issues
  - ✓ Sent only to MR-PCIM
    - Errors in new Multi-Root DLLPs

Base Errors

		Base	SR	MR
Physical	Receiver Error	RC	RC	Active RC & MR PCIM
Link	Bad TLP	RC	RC	Active RC & MR PCIM
	Bad DLLP	RC	RC	Active RC & MR PCIM
	Replay Timeout	RC	RC	Active RC & MR PCIM
	Repaly Num Rollover	RC	RC	Active RC & MR PCIM
	Data Link Layer Protocol Error	RC	RC	Active RC & MR PCIM
Transaction	Poisoned TLP Received	RC	RC	Affected RC
	ECRC Check Failed	RC	RC	Affected RC
	Unsupported Request	RC	RC	Affected RC
	Completion Timeout	RC	RC	Affected RC
	Completer Abort	RC	RC	Affected RC
	Unexpected Completion	RC	RC	Affected RC
	Receiver Overflow	RC	RC	Affected RC(s) & PCIM
	Flow Control Protocol Error	RC	RC	Affected RC(s) & PCIM
	Malformed TLP	RC	RC	Affected RC

New MR Errors

		Base	SR	MR
MR Errors	TLP received on VH in reset	n/a	n/a	MR PCIM
	Invalid VH in FC DLLP	n/a	n/a	MR PCIM
	FC DLLP for VH in Reset	n/a	n/a	MR PCIM
	Invalid VH Group in Reset DLLP	n/a	n/a	MR PCIM
	Out of range Assert bit set in Reset DLLP	n/a	n/a	MR PCIM
	TLP Prefix VL not valid for VH	n/a	n/a	MR PCIM
	TLP Prefix with Global Key Mismatch	n/a	n/a	Active RC & MR PCIM
	Invalid TLP Prefix	n/a	n/a	Active RC & MR PCIM

Virtualizing PCI Express Interrupts

Interrupt Goals

- Provide one or more vectors per VF
  - ✓ scalability/performance
- Identify the interrupt source
  - ✓ for correct operation
  - ✓ for efficiency
- Limit VI work
  - ✓ reduce the need to maintain state
- Compatibility with existing software models
  - ✓ Work with existing Guest OS / VM software
  - ✓ No requirement to change driver model

Base Specification

PCI Express Interrupt Support

- The PCI Express interrupt model supports two mechanisms:
  - ✓ Legacy INTx emulation
  - ✓ Message Signaled Interrupt (MSI/MSI-X)
- The rational of dual approach include:
  - ✓ Compatibility with existing PCI Software Models
  - ✓ Direct support for boot devices
  - ✓ Easier End of Life (EOL) for INTx legacy mechanism.
- MSI, MSI-X, or both is required for native PCI Express endpoint devices

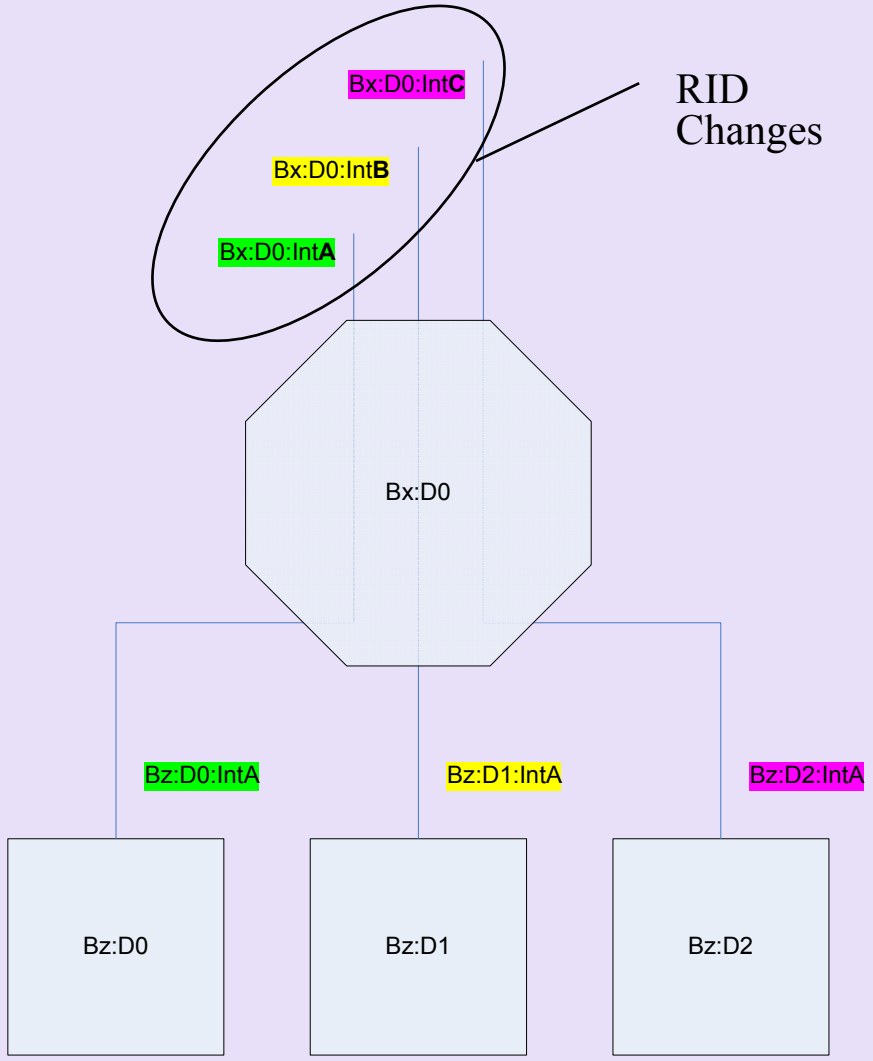
Legacy INTx

- Level triggered mechanism
- Switches collect and combine INTx
  - ✓ Switch uses is own RID when forwarding message to upstream port.
  - ✓ INTx Message changes as it passes through switches.

    Bz:D0:IntA -> Bx:D0:IntA

    Bz:D1:IntA -> Bx:D0:IntB

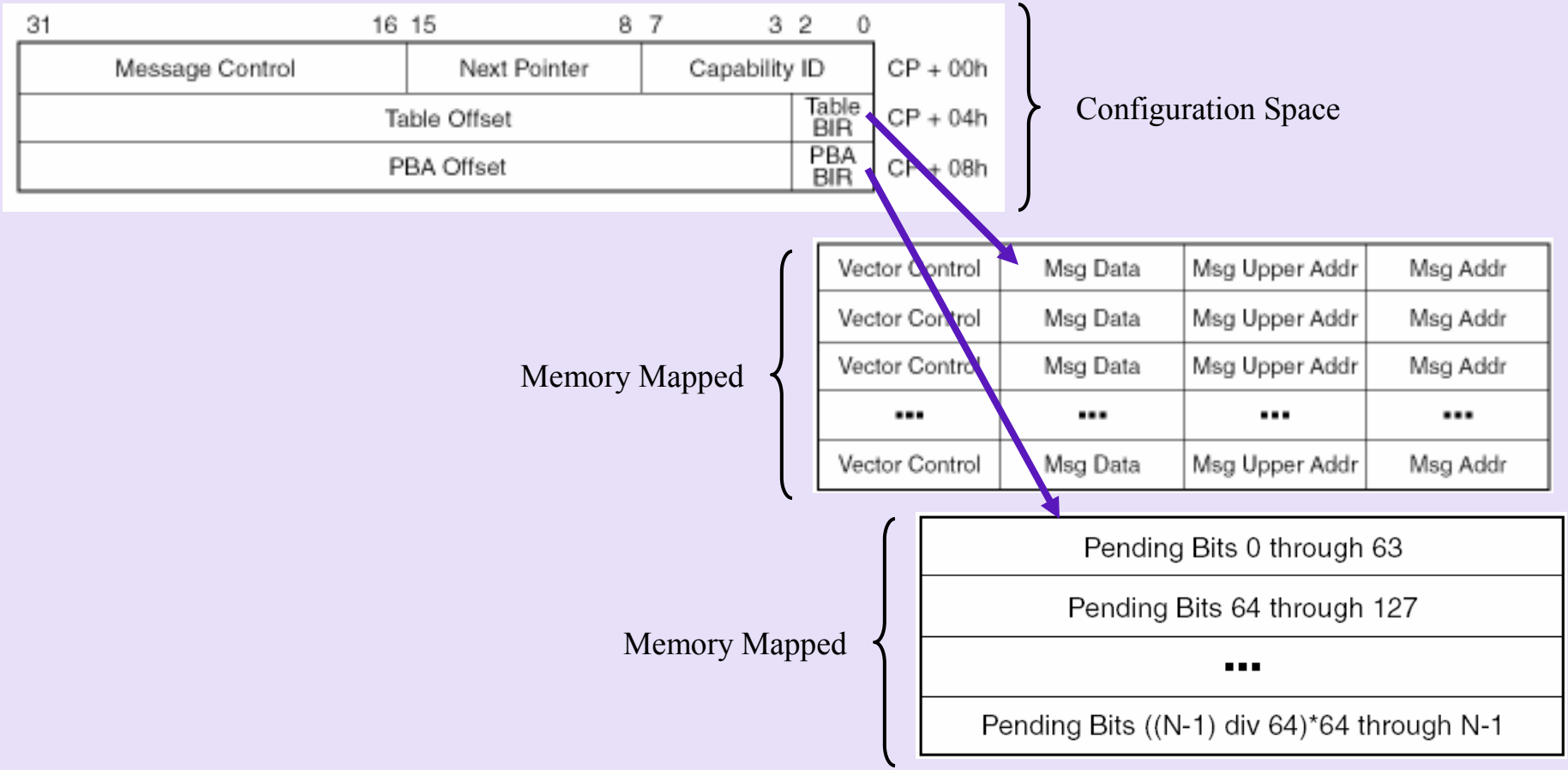
    Bz:D2:IntA -> Bx:D0:IntC
- Source of interrupt not part of INTx message



Message Signaled Interrupt

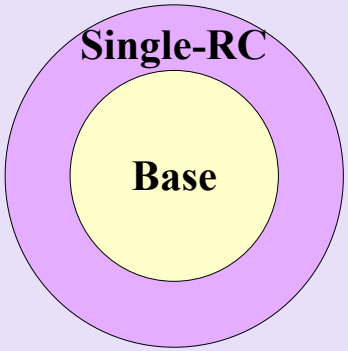
- MSI/MSI-X interrupt support required for PCIe devices
- Delivers interrupts via memory write transactions
  - ✓ Edge-triggered mechanism
  - ✓ 64-bit Message Address version of MSI is required.
- MSI and MSI-X have Capability Structures in Configuration Space.
- MSI-X requires MMIO region
  - ✓ MSI-X tables and Pending Bit Array are located in MMIO space
  - ✓ Offset relative to a BAR
  - ✓ MSI-X Table is recommended to be in its own 4KB page.

MSI-X Capability



Legacy INTx Virtualization

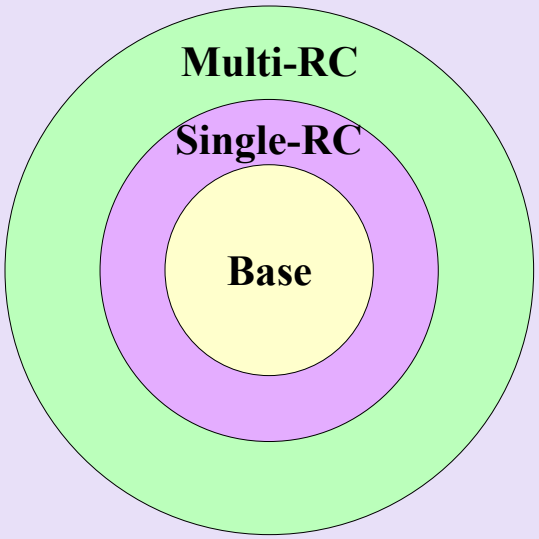
Single Root INTx Routing

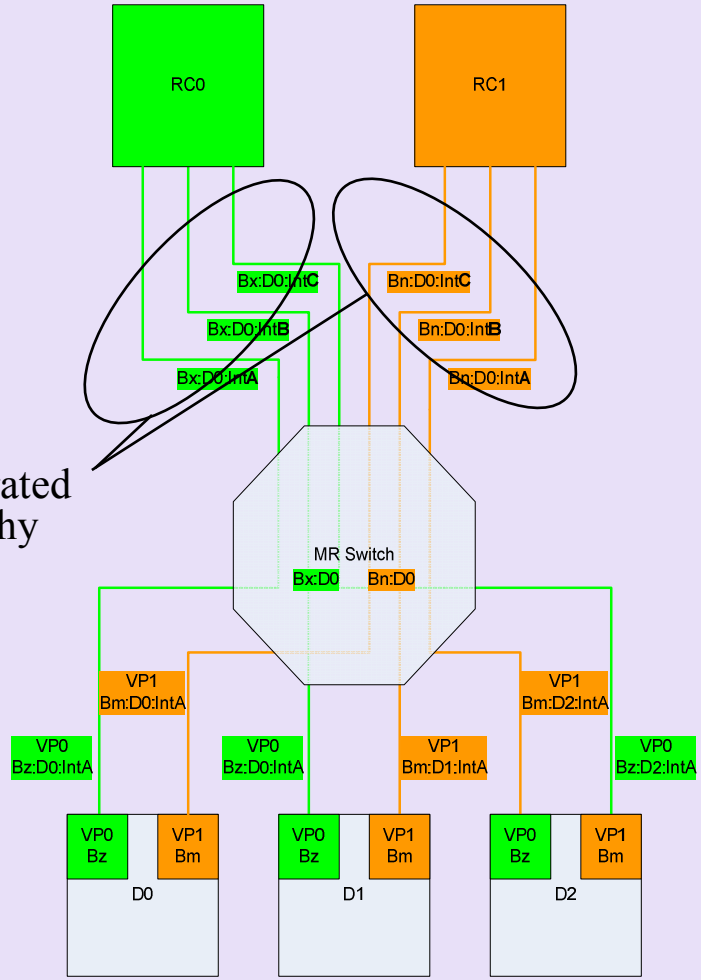


- No changes on the Wire
  - ✓ INTx delivered to interrupt controller
- Source device unknown
  - ✓ Intervening switches take ownership
- VI involved in interrupt delivery
  - ✓ Similar to OS role for a shared interrupt
  - ✓ VI calls SIs that share that Interrupt line
- PFs may generate INTx
- VFs may not generate INTx
  - ✓ VI may emulate INTx mechanisms

Multi-Root INTx Routing

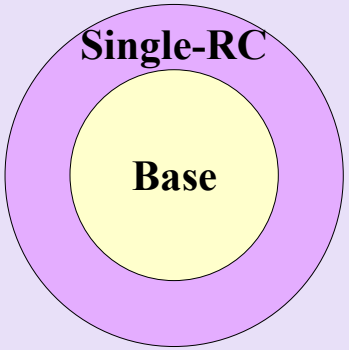
- Virtual Hierarchy Identifier
  - ✓ Applied at endpoint
  - ✓ Used by switch to route TLP
- INTx routes with a VH
  - ✓ Switch collects and combines per VH
  - ✓ SR rules apply within a VH





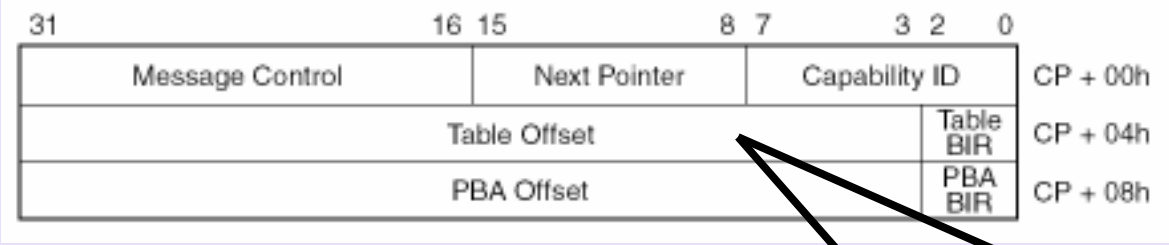
MSI and MSI-X Virtualization

Single Root MSI/MSI-X



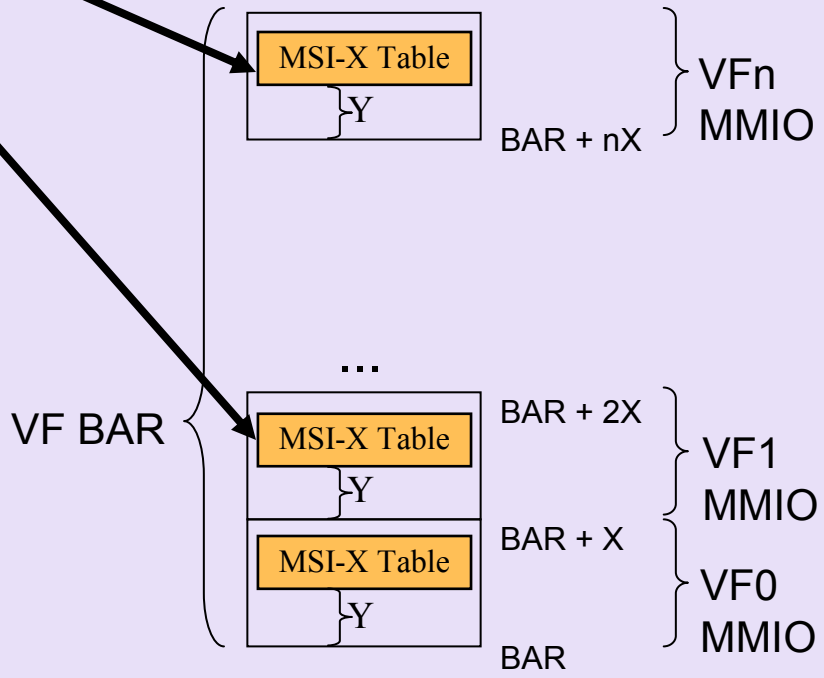
- No changes on the Wire
  - ✓ Memory mapped routing applies
- Source Identification Retained
  - ✓ TLP includes Requester ID
  - ✓ RID preserved throughout hierarchy
  - ✓ Can be used to route to the correct SI
- MSI, MSI-X or both required for PF and VF

Access to MSI-X Table



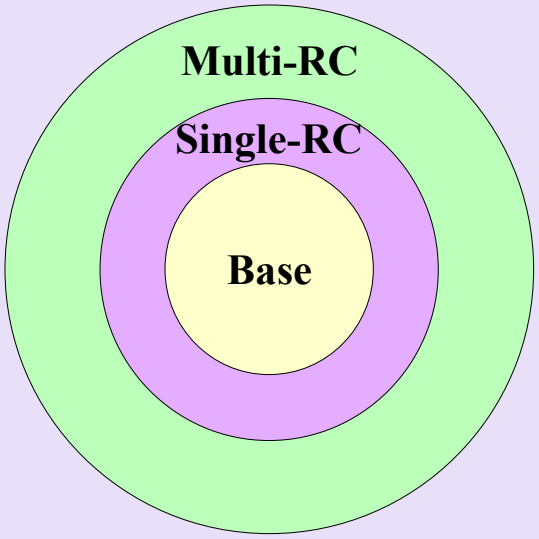
- BIR points to VF BAR
- Offset is relative to VF Base Address

- MSI-X Table / PBA
  - ✓ Replicated per VF
  - ✓ Located relative to VF MMIO regions



X is VF offset
Y is Table Offset

Multi Root MSI/MSI-X



- TLP routed within a Virtual Hierarchy
  - ✓ VH Tag is part of routing mechanism
- SR rules apply within the VH
- Source Identification Retained
  - ✓ TLP includes Requester ID
  - ✓ RID preserved throughout hierarchy

Virtualizing other Events

Events

- Event signaling in PCIe is performed via Message TLPs
  - ✓ INTx, Error, PME, Hot Plug
- Routed based on Message Routing code
  - 000 Route to Root
  - 001 Route by Address
  - 010 Routed by ID
  - 011 Broadcast from Root
  - 100 Local – Terminate at Receiver
  - 101 Gather and Route to Root
  - 110-111 Reserved
- SR Message Routing retains current behavior
- MR Message Routing stays within a Virtual Hierarchy
  - ✓ SR rules apply within the VH


PCI
SIG®