
PCI
SIG®

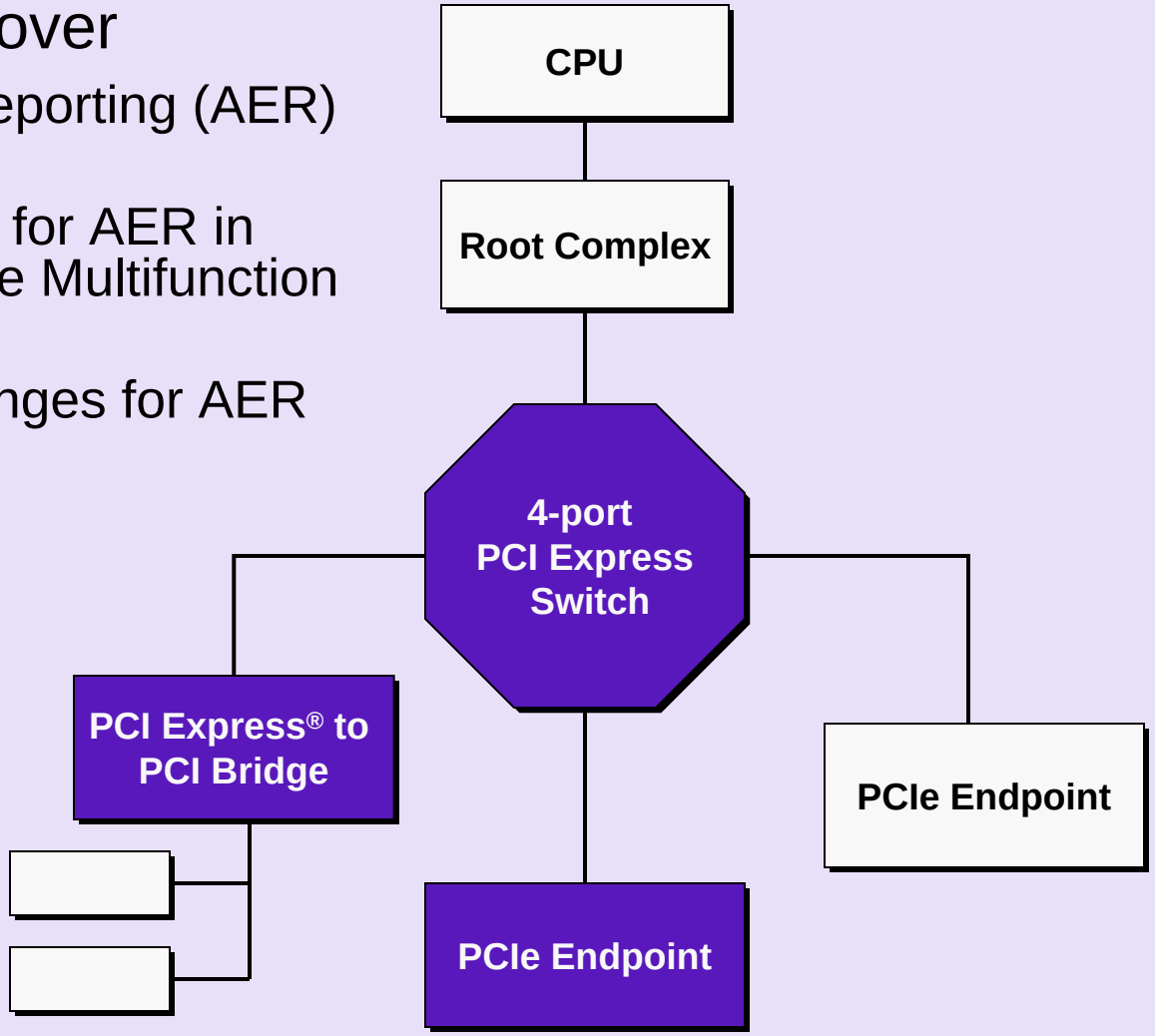
Verification of Advanced Error Handling Architecture

Sumit Das

(sumit@ti.com)

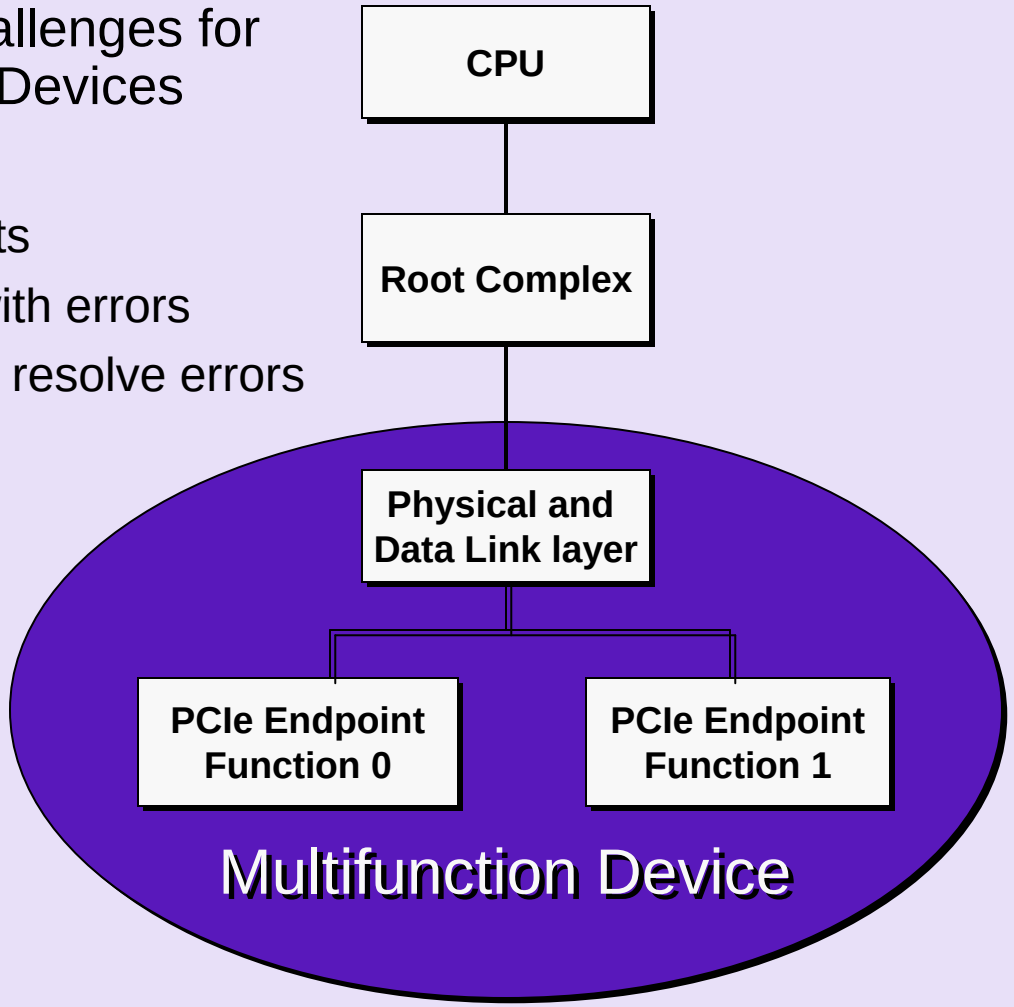
Purpose

- Presentation will cover
  - ✓ Advanced Error Reporting (AER) Architecture
  - ✓ Design challenges for AER in PCIe® Switch, PCIe Multifunction Endpoint
  - ✓ Verification Challenges for AER



Purpose (cont.)

- Design and Verification challenges for AER in PCIe Multifunction Devices
- AER
  - ✓ Stores data on error events
  - ✓ Stores headers of TLPs with errors
  - ✓ Gives software visibility to resolve errors



Outline

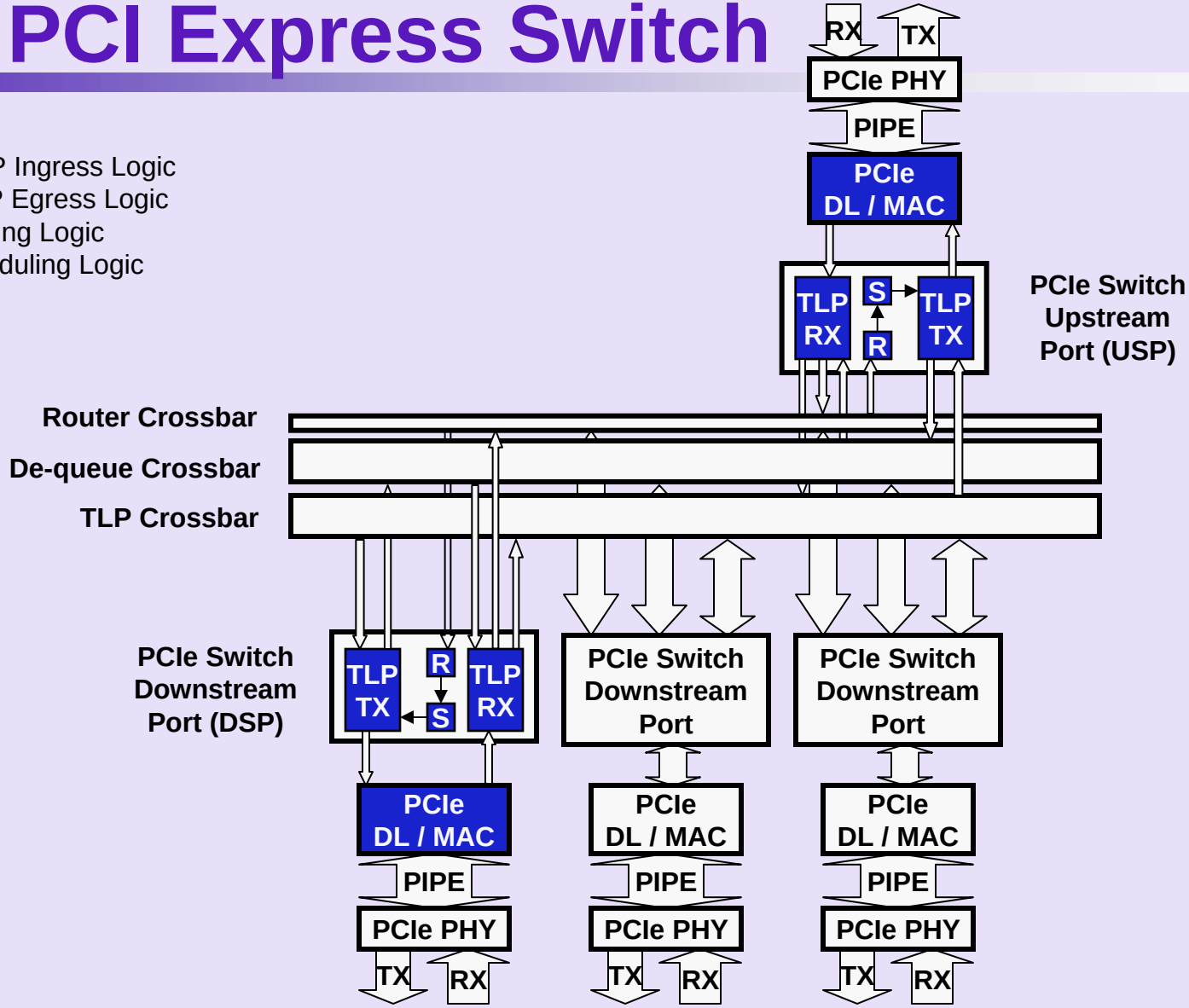
- Overview: AER, Switch, Multifunction Device
  - AER: Theory and Implementation
  - Example: Switch Error Reporting
  - Example: Multifunction Error Reporting
  - Conclusion

AER and PCIe Device Architecture

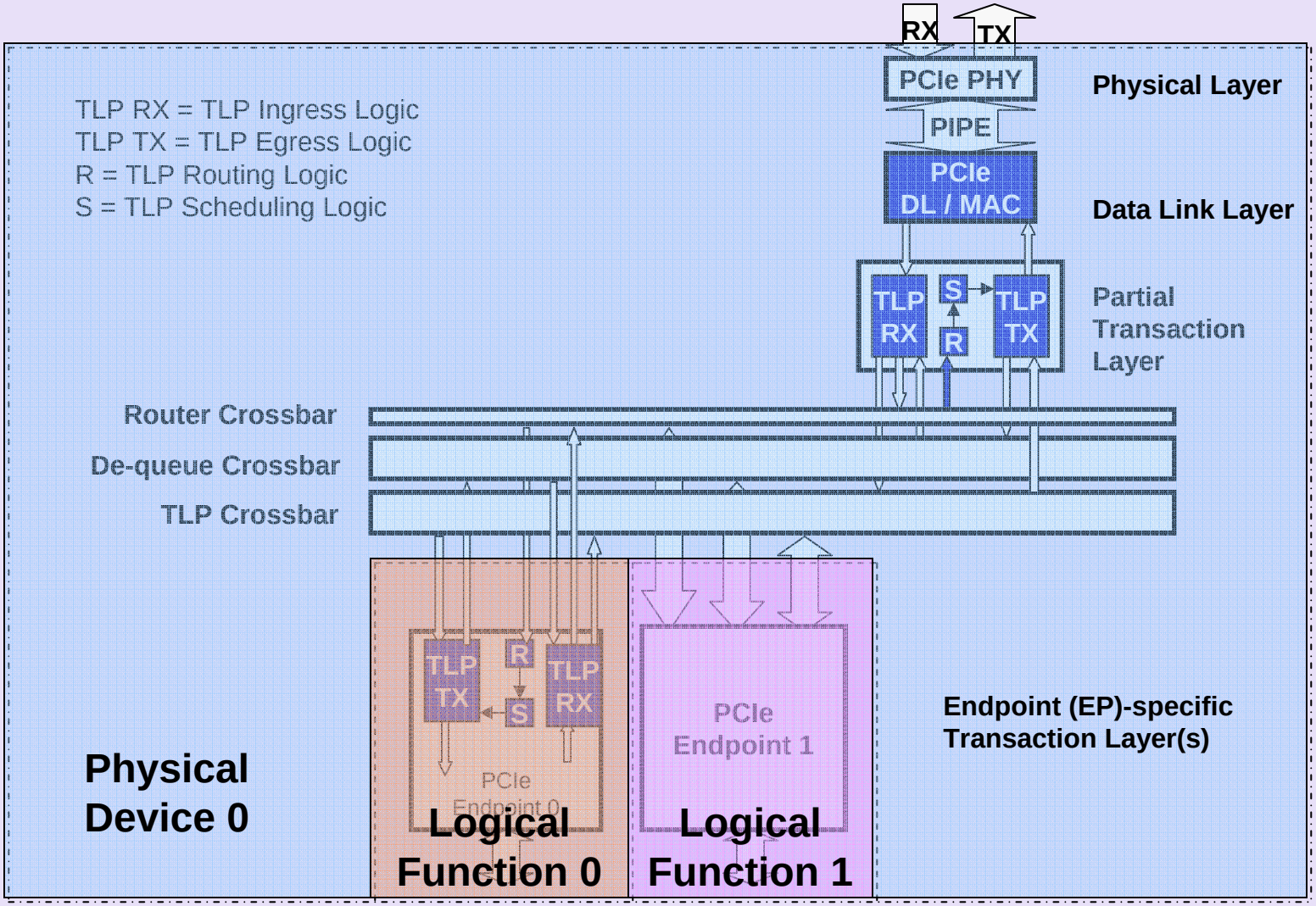
- PCI Express supports
  - ✓ Baseline Error Reporting
    - Generic recording of error events
  - ✓ Advanced Error Reporting
    - More robust than baseline error reporting
    - Differentiates errors as correctable and uncorrectable
    - Detailed information recorded for each error
    - Operating system may use this information to take corrective action

Example: PCI Express Switch

TLP RX = TLP Ingress Logic
TLP TX = TLP Egress Logic
R = TLP Routing Logic
S = TLP Scheduling Logic



Example:
PCIe Multifunction Device



Error Detecting Agent Architecture

- PCI Express devices need to detect
  - ✓ Physical layer errors
  - ✓ Data link layer errors
  - ✓ Transaction layer errors
    - Large variety of errors in TL compared with PL and DL
- Prioritize multiple errors in a single TLP

Error Detecting Agent Architecture (cont.)

- Header logging
  - ✓ TLP header is recorded to memory
  - ✓ Software may read header to determine corrective action
- First error pointer update
  - ✓ 5 bit register field
  - ✓ Keeps track of uncorrectable error bit
- Locking scheme for header updates
- Prevention of error pollution
  - ✓ Report only one error per TLP

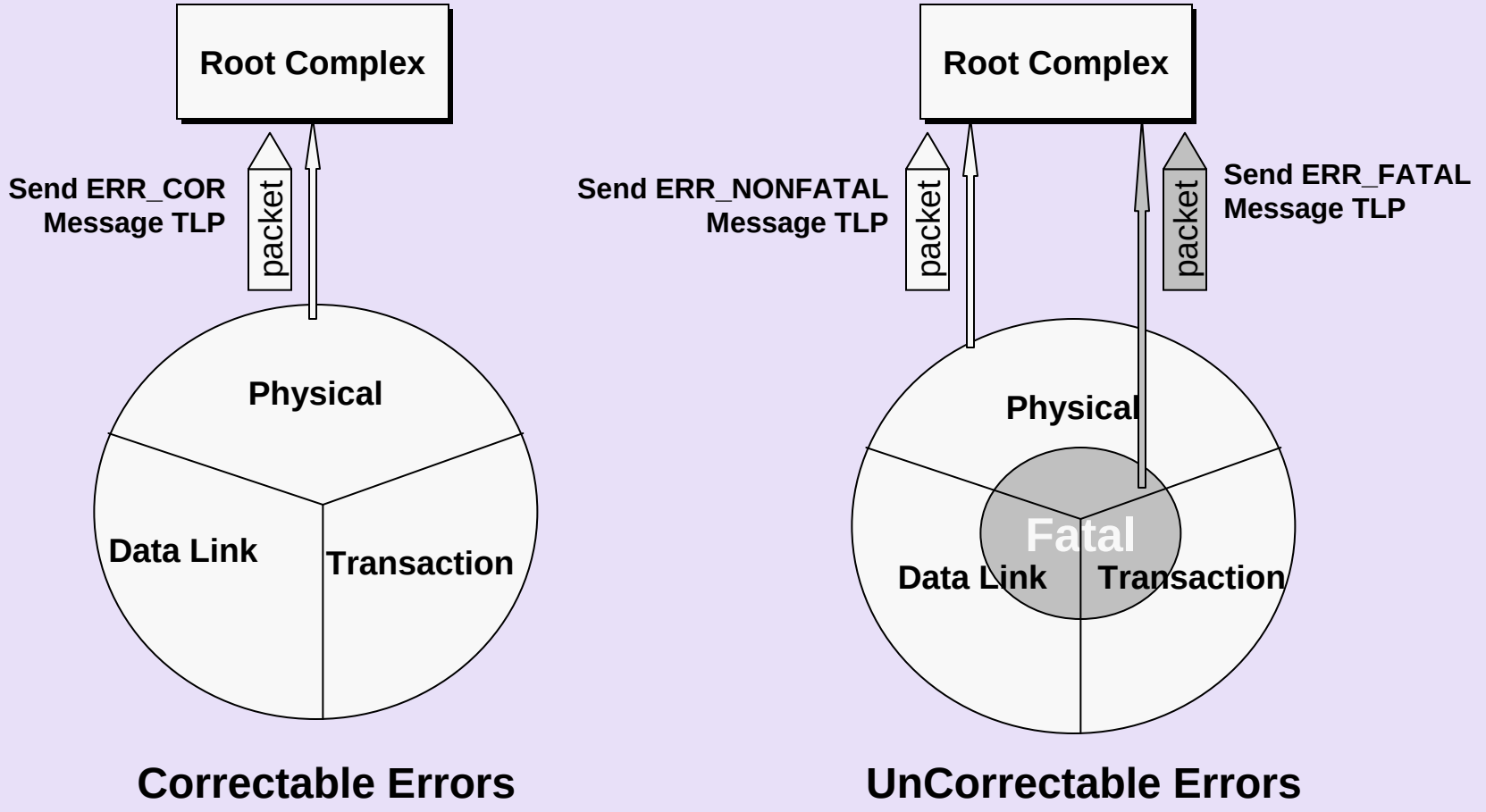
Distributed Error Detection Logic

- Switch
  - ✓ Errors are detected in multiple places
  - ✓ Each error will only be reported in one place
  - ✓ One error reporting mechanism per port
- Multifunction device
  - ✓ Errors are detected in multiple places
  - ✓ Some errors reported in a single function
  - ✓ Some errors reported in all functions
  - ✓ One error reporting mechanism per function

Outline

- Overview: AER, Switch, Multifunction Endpoint
- AER: Theory and Implementation
- Example: Switch Error Reporting
- Example: Multifunction Error Reporting
- Conclusion

PCI Express Error List



PCI Express Error Matrix

ANFE = Advisory NonFatal Error, **NFE** = NonFatal Error , **FE** = Fatal Error

ERROR TYPE	NFE	FE	ANFE	Switch USP	Switch DSP	MFUNC EP
Unsupported Request						
Address based window	x	x	x	x	x	x
ID based window	x					
Unsupported Message Request	x					
Unsupported Function Number	x					
Unsupported Downstream Port Numbers	x					

PCI Express Error Matrix (cont.)

ERROR TYPE	NFE	FE	ANFE	USP	DSP	MFUNCEP
Byte Enable error – various combination of first dw byte enable and last dw byte enable and payload length		x				

- Multiple combinations of errors and simulation scenarios
- Different configuration settings to control AER

PCI Express Error Register Bit Modeling

- Challenges for verifying AER status and control bits in a switch, bridge, or multifunction device
  - ✓ Variables:
    - PCIe baseline register control bits
    - PCIe AER register control bits
    - Role-based error reporting
    - Type of detected error
  - ✓ Many variables = complex functionality
  - ✓ Bit types:
    - Read/write
    - Write 1 to clear
    - Read only
    - Read only, hardware-updatable
  - ✓ Divide and conquer:
    - One RTL module per bit type
    - Verify each bit type
    - Verify the source of the error condition
    - Verify propagation of error events for each instance

Switch Virtual PCI Bridge Error Handling

- TLP handling rules apply at the ingress side of the Switch
  - ✓ Error Message forwarding rules
    - System error reporting enable
    - Device control register
      - Correctable error reporting enable
      - Uncorrectable error reporting enable
      - Unsupported Request (UR) reporting enable
- If upstream port detects error:
  - ✓ Error message generated when upstream port is configured to send the message

Virtual PCI Bridge Error Handling (cont.)

- If downstream port detects error:
  - ✓ Error message must flow through two virtual bridges
    - One at the downstream port
    - One at the upstream port
  - ✓ Error message will only be generated when *both* bridges are enabled to send the message.

PCIe Multifunction Error Reporting

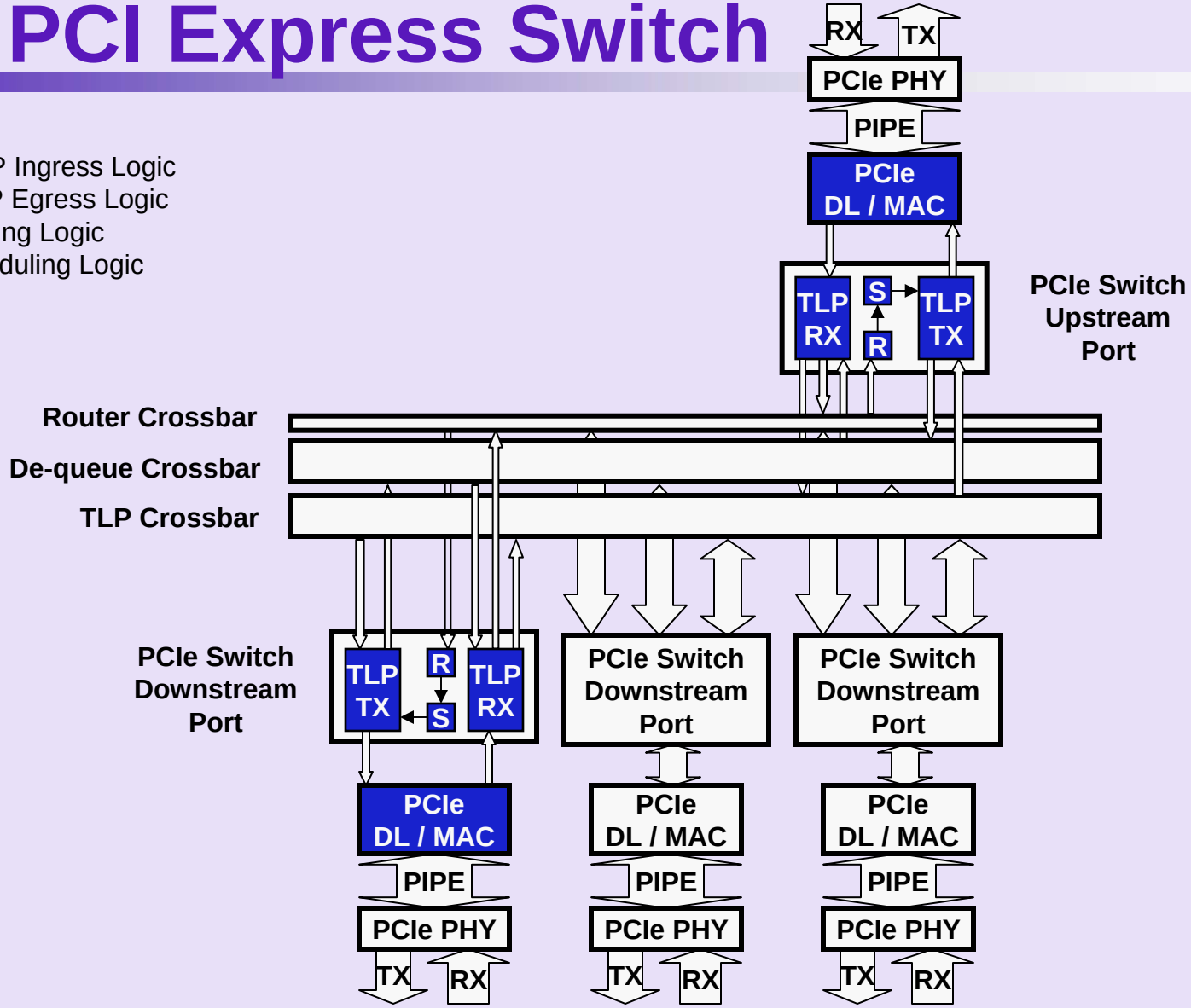
- Errors detected in the shared logic must be reported by *all* of the functions in the device:
  - ✓ All Physical Layer Errors
  - ✓ All Data Link Layer Errors
  - ✓ Some Transaction Layer Errors
    - Malformed TLP
    - Receiver Overflow
    - Request targeting unimplemented function
    - …

Outline

- Overview: AER, Switch, Multifunction Endpoint
- AER: Theory and Implementation
- Example: Switch Error Reporting
- Example: Multifunction Error Reporting
- Conclusion

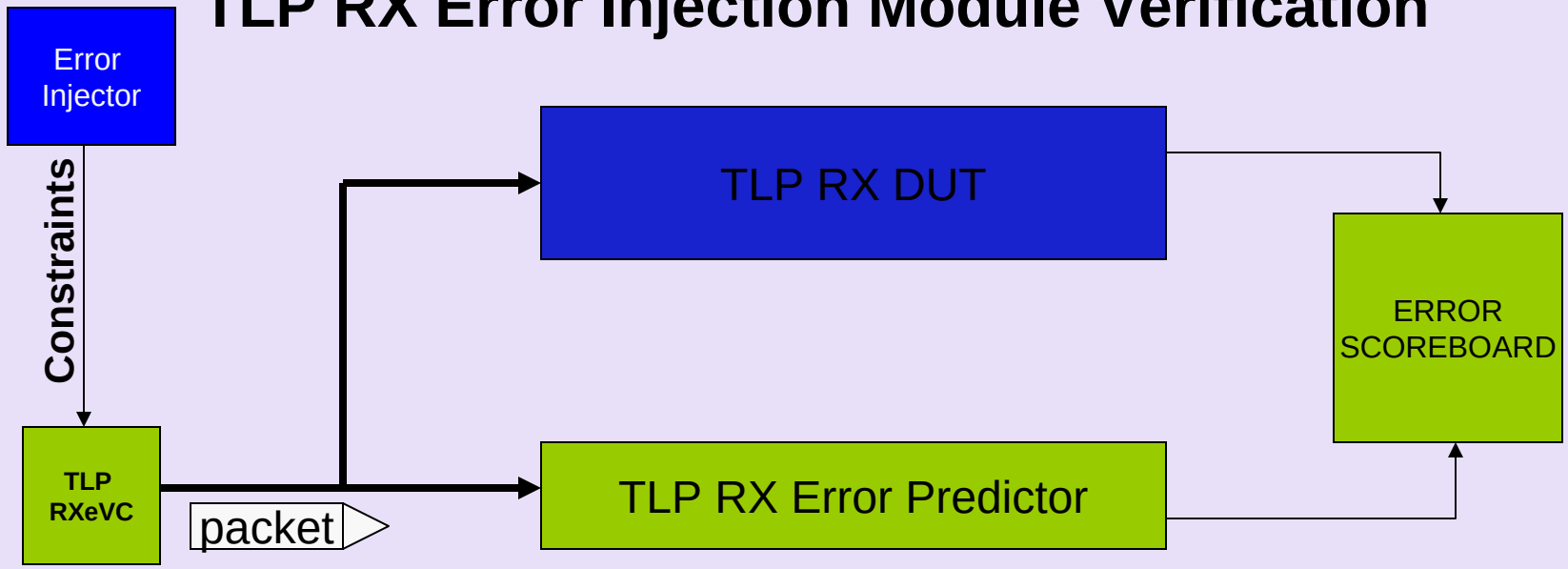
Example : PCI Express Switch

TLP RX = TLP Ingress Logic
TLP TX = TLP Egress Logic
R = TLP Routing Logic
S = TLP Scheduling Logic



TLP Error Detector

TLP RX Error Injection Module Verification



Error Injection Strategy

- Generate error cases in the testbench
- Constraint-based error injection at block- and chip-level
  - ✓ Multiple errors introduced by selecting enumerated types in constrained-random and directed fashion
  - ✓ Support for error injection at PL, DL and TL levels
  - ✓ Each can be turned on or off separately
- Constrainable message verbosity
  - ✓ Few messages for regression testing
  - ✓ More messages for single-test debug

Verification Difficulties

- Errors are detected in many places
  - ✓ Need to trace back to find source of error
- Different reporting paths are used depending on the mode of operation
- Multiple error reporting mechanisms
  - ✓ Each port in a switch has to report errors
  - ✓ Each function in a multifunction device has to report errors
- “So many paths, so little time”

“Divide and Conquer”

- Leverage the architecture
  - ✓ Hierarchical design
  - ✓ Modular, reusable blocks
- A module used multiple times needs to be verified only once
  - ✓ Fully verified at the module-level
  - ✓ Chip-level instances only need port connectivity check

Example: TLP RX module in switch

- One instance per port
- Buffers TLPs as they enter the port and holds them until they can be forwarded to the destination port
- Two modes of operation
  - ✓ Store and forward
  - ✓ Cut-through (TLP is entering as it leaves)

Example: TLP RX module in switch (cont.)

- Error checking
  - ✓ Malformed header checks
    - All required checks are performed
    - Some optional checks are performed
  - ✓ Data length error checks
    - More data then expected
    - Less data then expected
  - ✓ ECRC check
    - Only check when the digest bit is set in the header

Example: TLP RX module in switch (cont.)

- Error reporting
  - ✓ Pass information to error processing logic
    - TLP header
    - Tag with error type of highest priority
    - All errors reported through single bus to error logic
  - ✓ Only one report per TLP
  - ✓ Cut-through TLP
    - Save potential error data when TLP starts to exit TLP RX module
    - Update error data if an error occurs while the TLP is passing through the TLP RX module
    - Discard potential error data if TLP is nullified or completes without an error

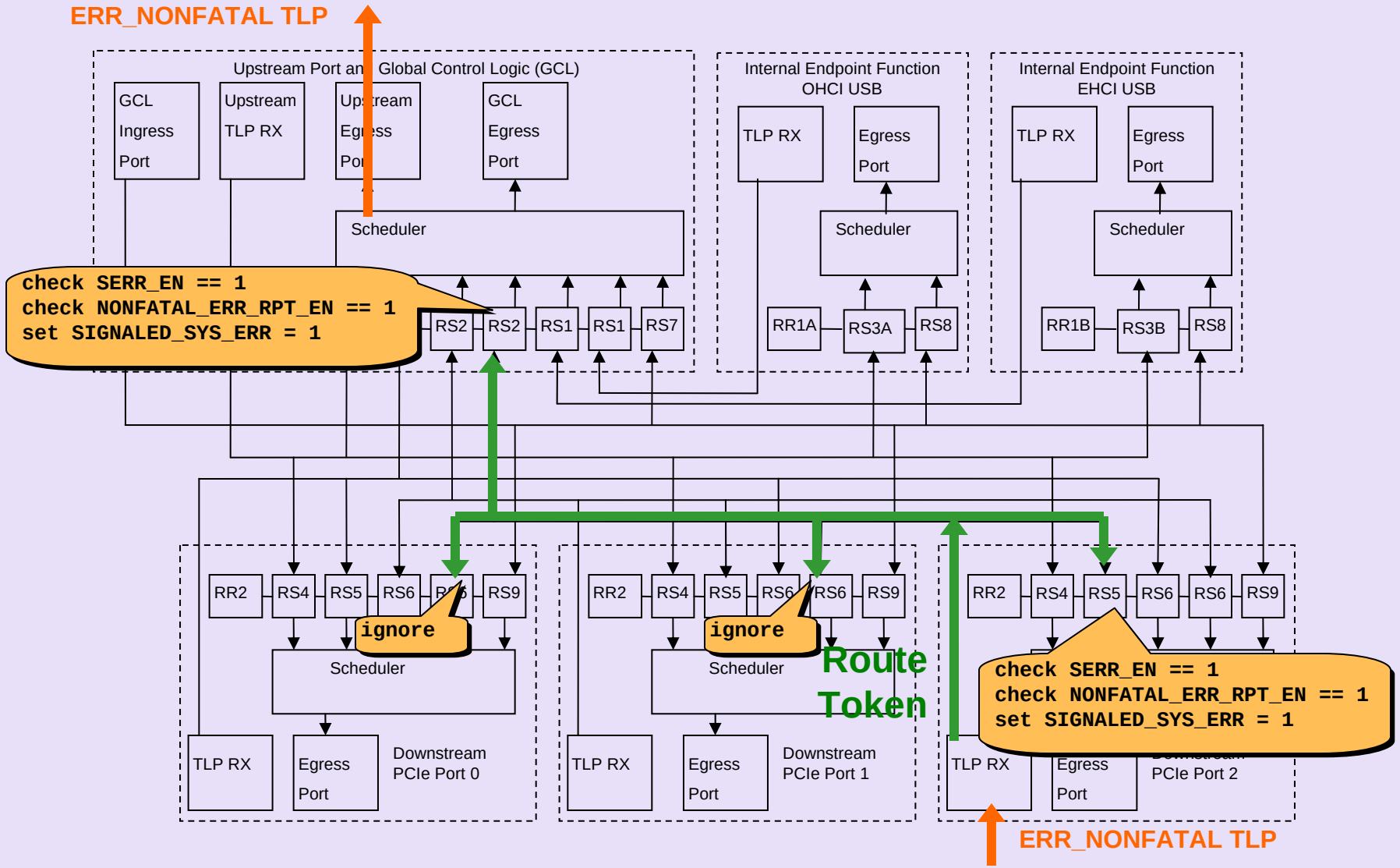
Example: TLP RX Module Verification

- Verification was done at the module-level
  - ✓ Easier to create the stimulus to test the many paths through the module
    - Create stimulus at module inputs
    - Verify error data or lack of error data at the error output bus
  - ✓ Smaller model with faster run time
  - ✓ Less logic under test, therefore easier to debug problems

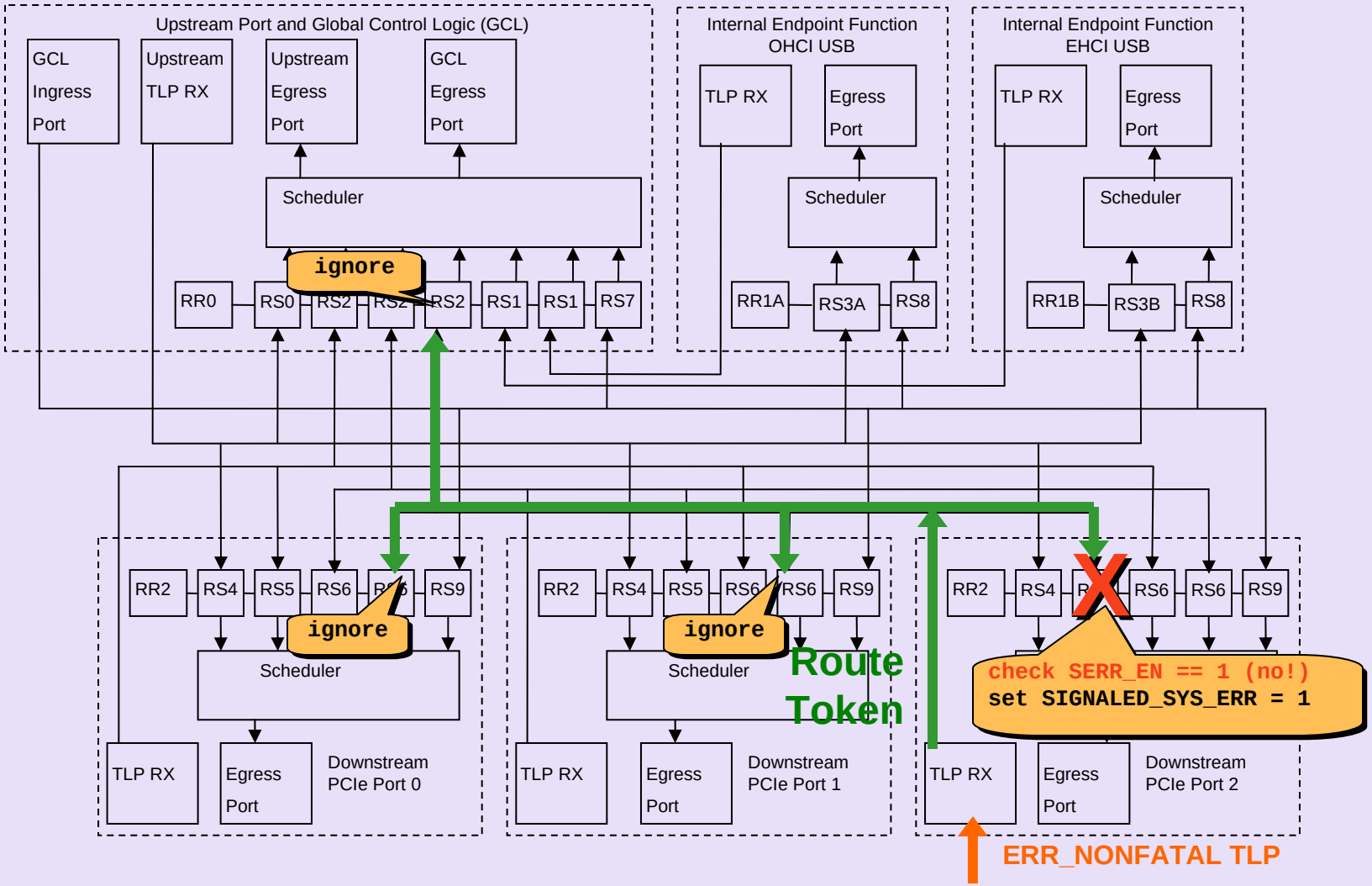
Error Handling in the Routers

- Routing implementation does not look like programming model
  - ✓ Distributed routing modules
  - ✓ Distributed configuration space
  - ✓ Therefore: Distributed error handling
- Example: Violation of Max Payload Size
  - ✓ Router senses violation on egress-side of chip
  - ✓ Router informs the error handling logic
- Example: Error Message TLPs
  - ✓ Error messages “bubble up” through the chip
  - ✓ May be gated at each virtual internal bridge
  - ✓ May trigger status bits to be updated

Routing example: Msg ERR_NONFATAL



Routing example (2): Msg ERR_NONFATAL

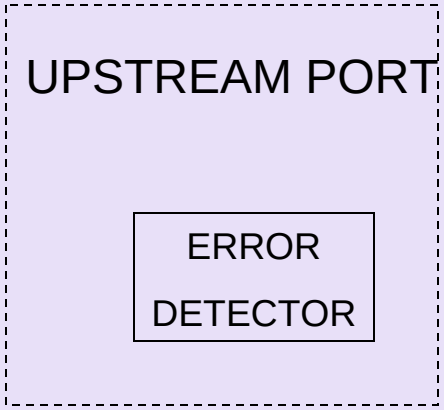


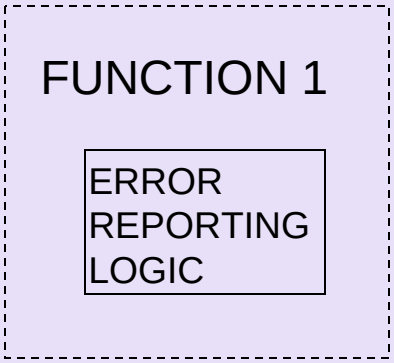
Outline

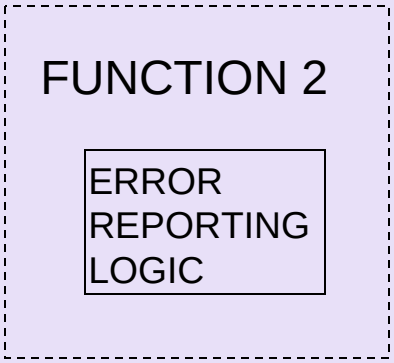
- Overview: AER, Switch, Multifunction Endpoint
- AER: Theory and Implementation
- Example: Switch Error Reporting
- Example: Multifunction Error Reporting
- Conclusion

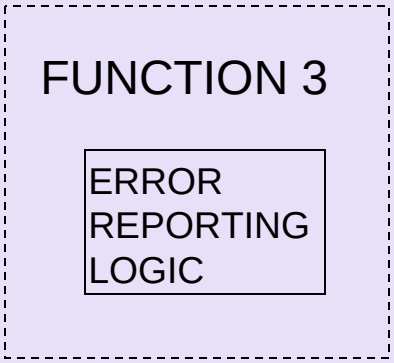
Error Forwarding in Multifunction Device

- Upstream port receives TLPs with various errors
- Some errors need to be reported by all functions within the same physical device
- Problem: How to propagate a single error across multiple functions?



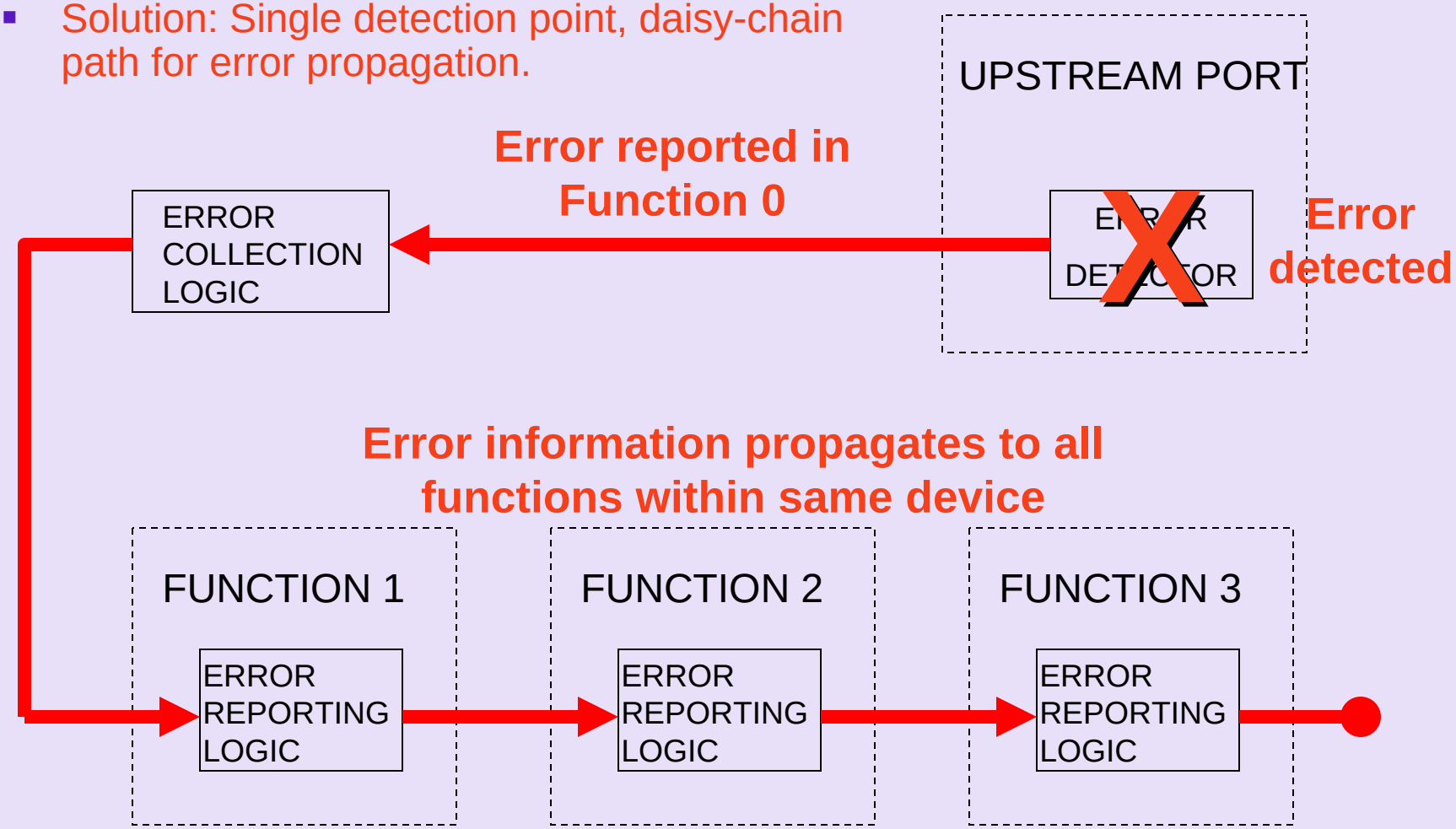






Error Forwarding in Multifunction Device (cont.)

- Solution: Single detection point, daisy-chain path for error propagation.



Multifunction Advanced Error Checking

- Block-level error testing
  - ✓ Verify detection of various completion packet errors
- Chip-level error testing
  - ✓ Error aggregation logic for multi function device
  - ✓ Verify reporting logic for all functions in case of
    - All physical layer errors
    - All data link layer errors
    - Transaction layer errors
      - Malformed TLP
      - Receiver Overflow
      - Request targeting unimplemented function
      - …

Outline

- Overview: AER, Switch, Multifunction Endpoint
- AER: Theory and Implementation
- Example: Switch Error Reporting
- Example: Multifunction Error Reporting
- Conclusion

Conclusion

- Define the task
  - ✓ Verification matrix for various error conditions detected by device
  - ✓ Decide which optional checks will be implemented
- Don’t just design. Design for Verification.
  - ✓ Minimize types of unique logic blocks
- Modular approach
  - ✓ “Divide and conquer”
  - ✓ Use module-level verification to your advantage

Conclusion (cont.)

- Module-level AER verification
  - ✓ Focus on block-level internal error detection and reporting
  - ✓ Rigorous simulation of all possible combinations
- Chip-level AER verification
  - ✓ Verify connections of the modules
  - ✓ Verify module interactions
    - End-to-end detection and reporting
    - Example: Error event on downstream port causes error message transmission on upstream port
    - No error pollution!
  - ✓ Special cases
    - Error status bits on different reset domains
    - Layered bridges cause error message filtering

About the Authors

- **Sumit Das (sumit@ti.com)**
  - ✓ Design Engineer for PCI Express Switch , Multifunction Bridge , PCIE-PCI Bridge .
- **Henry Angulo, SMTS (h-angulo@ti.com)**
  - ✓ Design Verification Lead for PCI Express Switch and Multifunction Bridge
- **Asad Khan (a-khan1@ti.com)**
  - ✓ Design Verification Engineer for PCI Express Switch, FireWire, PCI
- **Scott Morrison (scott@ti.com)**
  - ✓ Design Verification Engineer for PCI Express Switch, USB PHY, SERDES
- **Roy Wojciechowski, MGTS (roywojciechowski@ti.com)**
  - ✓ Design Engineer for PCI Express Switch, FireWire, PCI

Thank you for attending the
PCI-SIG Developers Conference 2007.

For more information please go to
www.pcisig.com


PCI
SIG®