Bullet-Proofing PCIe in Enterprise Storage SoCs with RAS features

Michael Fernandez, Sr. FAE, PLDA

Flash Memory Summit 2018
Santa Clara, CA

Agenda

- What is RAS(M)?
- PCIe RAS features
  - What's in the Spec. and what's not
  - Limitations
- Case studies
  - Problem encountered
  - Solution provided
  - Generalization
- Wrap-up

RAS(M) for Enterprise Storage

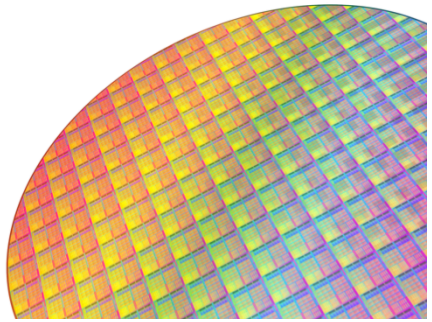
- **R**eliability
  - My storage device should not fail
- **A**vailability
  - If my storage device fails, the system should keep working
- **S**erviceability
  - I should be able to fix/replace my failed device easily and rapidly
- **M**anageability
  - I should be able to control, supervise, monitor my device constantly

RAS Features in PCIe Protocol

- ECRC, LCRC, ACK/NACK
  - Ensures what is sent is what is received
- ACK/NACK timeouts
  - Ensures link partner is alive
- Various timeout counters, LTSSM recovery mechanisms
  - Ensures device does not get locked
- Advanced Error Reporting (AER)
  - Provides more detailed error detection and reporting

Extended RAS Features for PCIe

- Accepted though not part of Spec
  - ECC for memories
  - Parity for data path
- Additional level of reliability but
  - Typically only 1-bit errors are correctable (ECC)
  - Costly, ex. 32-bit required to protect 256-bit data pat



RAS Limitations in PCIe Protocol

- Focuses on link, Tx to Rx data path, and payload
- Does not cover
  - Non compliant link partner, ex. DLLP latency, LTSSM timer
  - Performance issues, ex. retries due to poor channel quality, credit starving
  - Application logic bad/non-compliant behavior, ex Error injection

The Need for More R, A, S (and M)

- Shrinking process nodes, smaller transistors
  - Increased risk of errors due to external disturbances (EMI, heat, power supplies, etc.)
- Increasing PCIe speeds
  - Increased risk of errors due to tighter timing budgets
- Growing number of PCIe components and devices
  - Increased interoperability issues

Reliability: Non-Compliant Link Partner

Problem Example:

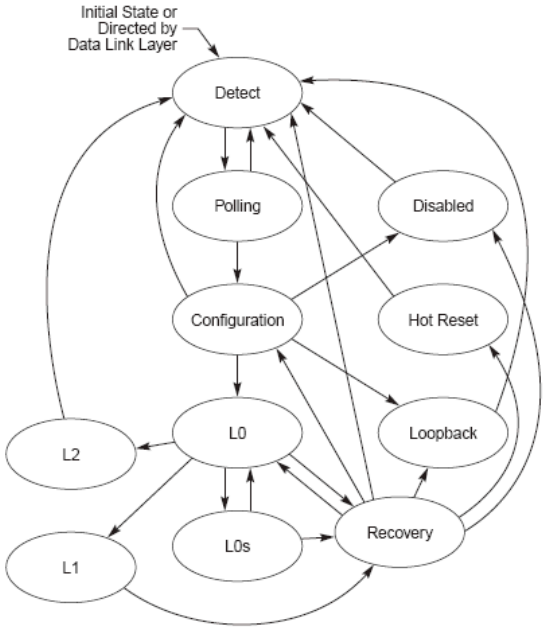
All timers in the PCIe specification are minus 0 seconds and plus 50% unless explicitly stated otherwise.
In Equalization Phase, PHY Figure of Merit (FOM) result for a specific preset may exceed this timer causing a link quality issue or downgrade to GEN1.

Proposed Solution:

Use plus 50% margin for all timer in LTSSM to provide FOM for multiple Preset to achieve correct BER

Generalization:

- Dynamically Programmable LTSSM timers
  - Allow values that extend beyond Spec.
- But, over 30 timers in LTSSM
  - Need to select which make sense



Reliability: Non-Compliant Link Partner (2)

Problem Example:

Link Partner ACK latency is out of PCIe spec. causing unexpected replays impacting performance.

Proposed Solution:

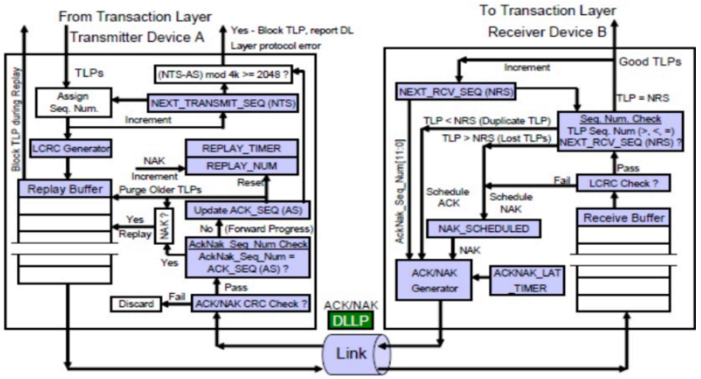
Increase ACK/NAK timeout values to reduce unnecessary replays.

Generalization:

- Dynamically programmable ACK/NACK and Replay timers

Limitation:

- Size of Replay Buffer



Reliability: Improving Tolerance to Errors

Problem Example:

How your chip is behaving if the Link Partner is injecting errors?

Malformed TLPs can be sent and cause credit leakage when ECC or parity errors are detected while transmitting.

Proposed Solution:

Inject Errors: Allows controlled testing through registers

Detect Errors: Provides triggers, logging and interface notification

Prevents and nullified transmit TLPs and release associated credits when ECC/Parity errors are detected

Generalization:

- Implement error injection and detection mechanism & interface
- Enable error injection in TLPs, DLLPs, OSs
- Pandora's box = scope definition required

Serviceability: PHY/PCS Monitoring

Problem Examples:

No Link UP: Link training issue (receiver detect/ TSx received)

Proposed Solution:

Narrow Down the issue between PCIe hierarchy(RP/EP/SW) and within PCie device (PHY/PCS/MAC/AL)

- Monitoring the PIPE interface
- Errors Counters

Generalization:

- ❖ Provide snoop module for capturing/storing raw PIPE interface data.
- ❖ Provide to user an interface to trigger different type of errors (recovery …)

Serviceability: Quicker Debugging

Problem Example:

Frequent trips to Recovery state: Who initiated the recovery? What caused the recovery?

Error message received or sent: What type of error has been flagged? ex: Unsupported TLP: UR TLP received by the application

Proposed Solution:

Test interface with different level of errors and status information

- LTSSM probing interface
- RX TL probing interface with error flagging

Generalization:

- At minimal, add flags indicating which side of link initiated Recovery.
- Add interface for Recovery event logging/tracking.
- Add interface for RX/TX Path event logging/tracking.

Manageability: Performance Tuning

Problem Example:

PCIe link performances is shared between different packets types : TLPs and DLLPs.
DLLP scheduling may impact the overall performances (FC Update, ACK send too often or not).

Proposed Solution:

Configure priority level for each DLLP type (FC update timer and ACK timer)

Generalization:

- Enable different priority levels for FC updates
- Implement threshold and delay mechanisms

Manageability: Performance Tuning (2)

Problem Example:

PCIe specification provides low power mode when there is no activity on the link.
How can we save power during low activity?

Proposed Solution:

PCIe feature: during low traffic, link width and/or speed can be tuned to save power on both sides of the Link.
Reduce dynamically internal clocking.

Generalization:

- Autonomous Link speed/Width depending of the traffic.
- Dynamically adjust application layer clock.

Manageability: Measuring Performance

Problem Example:

How can you fine tune your chip performances without monitoring?
Application only has access to TLPs that its request/received.

Proposed Solution:

Internal counters or/and test interface to provide statistics of PCIe link usage

Generalization:

- Internal Module that measure in real time the overall performances, IDLE time, DLLP, TLP …

Wrap-up

- Implementing RAS(M) features in PCIe interface IP is crucial in today's SoCs and even more so in tomorrow's SoCs
- But RASM support is a Pandora's Box, and can have an impact on gate count, functionality and verification
  - Need to be carefully defined, architected, implemented, and tested
- PLDA continuously incorporates RASM features into its range of PCIe controller IP
  - Driven by customers in Storage, Networking, HPC, AI, and Automotive