Performance Impact of Correctable Errors on High Speed Buses

Daniel Ballegeer, David Blankenbeckler, Subhasish Chakrborty, Tal Israeli
Intel Corporation
Santa Clara, CA, USA
Emails: {dan.g.ballegeer, david.blankenbeckler, subhasish.chakraborty, tal.israeli}@intel.com

Abstract— **Modern high speed serial buses are generally required by specification to achieve a maximum bit error ratio. Are these requirements too restrictive? This paper will look at a series of studies on Peripheral Component Interconnect Express and Serial AT Attachment, investigating the impact of bit error ratio on bus performance. The results of these studies suggest that typical bit error ratio requirements may be conservative. The findings suggest that alternative bus performance specifications should be considered that would open new possibilities for design, validation and manufacturing test tradeoffs.**

Keywords-bit error ratio; BER; electrical validation; high speed interconnect; high speed bus; I/O.

I. INTRODUCTION

Modern high speed serial bus specifications generally have a requirement for maximum Bit Error Ratio (BER) [1][2][3][4]. In this context, bit error ratio is defined as the fraction of bits transmitted over the high speed interconnect that are interpreted incorrectly at the receiving device—i.e., a bit originally transmitted as a “1” is interpreted as a “0” or vice-versa. Table I summarizes these for a variety of buses: Third Generation Peripheral Component Interconnect Express (PCIe Gen 3), 10 Gigabit Ethernet, Serial AT Attachment (SATA), and Universal Serial Bus (USB). Note that there is no inherent need or expectation that each interface type has the same BER requirement, but the table illustrates that 10^{-12} is quite commonly used.

Many high speed buses such as the ones listed in Table I utilize error detection schemes such as a Cyclical Redundancy Check (CRC) at the receiving device to detect any signal integrity-induced bit errors that could have occurred over the interconnect. In such a scheme, in the event of a detected error, a request is sent to the transmitting device to send the data again (a retry). Ideally, a target BER level on an interconnect that employs a CRC check must take into account both the effectiveness of the CRC scheme with respect to the protected data packet size as well as the performance losses that result from error-induced retries on the bus. Although studies and publications on the effectiveness of CRC error detection have occurred for multiple decades [5][6], as far as the authors know, there have been few, if any, studies done on real world performance impact at various error rates. Some theoretical calculations of latency impact vs. error percentage have been presented [7], but this would not take into account other factors that interact with the error retries and contribute to the overall performance impact on a true workload. This paper will outline the results of several studies conducted to better understand the real world performance impact of increasing error rates beyond the specification level.

TABLE I. BER SPECIFICATIONS FOR SOME HIGH SPEED BUSES

Link	BER Spec
PCI Express Gen 3	10^{-12} [1]
10 Gigabit Ethernet	10^{-12} [2]
SATA 3.x	10^{-12} [3]
USB 3.x	10^{-12} [4]

These studies are interesting in that they provide some data justifying room for design tradeoffs. For example, there may be significant cost savings opportunities, trading off slight performance impact for lower cost material. Consider the example of a system design with long Peripheral Component Interconnect Express (PCIe) bus routing lengths. Instead of using more expensive low loss Printed Circuit Board (PCB) material, it may make sense to sacrifice error rate and realize a cost savings with standard FR4 PCBs. Likewise, there may be power reduction opportunities for low power devices, trading off performance for lower power operation, without sacrificing data integrity.

It is easy to show through either empirical measurements or theoretical arguments that the bus BER of a product is a distribution when measured across multiple instances of that product. Factors that induce this distribution include, among other things, the variations in the characteristics of the board interconnects, receiver circuitry, and transmitter circuitry. For example, in the voltage domain of the bus signal, these factors lead to a distribution of the electrical margin, V_m, where V_m represents the amount of voltage swing at the receiving device beyond the minimum required voltage detection threshold.

To simplify the example, consider an ideal case where there is no noise in the system when V_m is measured, i.e., V_m is the noise-free voltage signal margin. In a real system, bit errors result from noise adding or subtracting from this margin. In a zero-mean additive white Gaussian noise model such as that described in [8], the V_m distribution may be mapped into a BER distribution via the following relationship:

$$BER = \frac{1}{\sigma\sqrt{2\pi}} \int_{V_m}^{\infty} e^{-\frac{x^2}{2\sigma^2}} dx \, , \tag{1}$$

where σ is the standard deviation of the Gaussian noise.

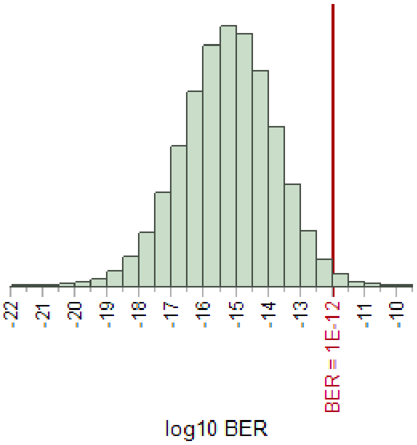

Figure 1. Conceptual illustration of bus BER distribution across different units

Fig. 1 is an example illustration of a BER distribution that could result from a Gaussian distribution of V_m. The specifications are a hard cut off at maximum BER limit, like 10^{-12}. In reality, the BER performance of every lane on every channel on every board is different. Is it acceptable that a small portion of lanes are slightly above the BER spec if the performance impact is negligible? Are these systems really considered bad if there is no noticeable performance impact to the end user? If it were acceptable to ship some portion of systems at a higher BER, there may be substantial benefit, such as the opportunity to reduce silicon test time requirements.

II. Performance Impact Experiment Overview

In order to measure the performance impact of BER levels above specification, four different high speed serial bus usage scenarios were studied:

- a PCIe Gen 3 bus used with a graphics add-in card
- a PCIe Gen 3 test add-in card utilized for easy measurement of data mismatch errors
- PCIe Gen 3 used as an interconnect between a CPU and Platform Control Hub (PCH)
- a Serial ATA (SATA) 6 Gb/s interconnect attached to a hard disk drive.

In all experiments, techniques were used to induce different BER levels on the link, either by

- changing the voltage or timing sampling at the receiving device to be offset with respect to the data eye center
- error injection at the receiver, or
- voltage swing attenuation at the transmitting device.

Then, with this induced BER present, performance benchmarks were run that specifically focused on the I/O being studied. In some cases, the BER was able to be monitored at the same time as the performance, whereas in other cases, BER had to be measured first in a loopback scheme before running the performance benchmark at the same settings. In all cases, experiments were re-run at least one time to confirm the performance results quoted.

III. Performance Data Collection and Results

A. *PCIe Gen 3 used with a graphics add-in card*

In the first experiment, the PCIe Gen 3 bus studied was the interconnect between a 3rd Generation Intel Core i7 Processor and a PCIe graphics Add-In Card (AIC). Four different high-performance graphics cards were included in the experiment, spanning three different vendors. Graphics card settings were set to produce maximum performance; future studies will also include studies with the scenario where hardware acceleration is turned off. Three different commercially available graphics-intensive benchmarks were run during the experiment: Codemasters Dirt 3, a graphics-intensive racing game; Unigine Heaven, a graphics-intensive benchmark designed to stress graphics AICs, and 3DMark Fire Strike, a real-time graphics rendering benchmark. In this experiment, the degradation of performance vs. BER was measured in both directions: in one set-up, the CPU receiver experienced the bit errors, and in a separate set of measurements, the graphics AIC receiver experienced the bit errors.

The CPU PCIe Receiver (Rx) circuitry had built-in validation test hooks that allowed changing the location of the sampling point in the time domain with respect to the data eye center. By offsetting the sampling point away from the data eye center, bit errors could be induced at the receiver.

In the first step, the BER vs. time sampling offset was established by sending a known random bit sequence out the CPU transmitter and receiving the same bit sequence at the CPU receiver. This was accomplished via the far-end digital loopback mode supported by all PCIe spec-compliant components. In this mode, the AIC received and interpreted the data transmitted by the CPU and then retransmitted the same data back to the CPU. Received data at the CPU was compared to the CPU transmitted data to detect the level of bit errors at each sampling offset point. Note as sampling offset moved closer to nominal data eye center, more transmitted bits were necessary to detect bit errors. BER vs. sampling offset slope was checked to ensure the relationship agreed with an additive white Gaussian noise model indicative of Random Jitter (RJ).

In the second step, the identical set of time sampling offset values were used in the same system setup, this time allowing the three benchmarks to run and measure performance. In this way, a correlation of performance vs. BER at the CPU receiver could be established.

To establish the relationship between performance impact of bit errors received by the Graphics PCIe Rx, a slightly different approach was used to induce bit errors: the CPU transmit voltage swing was reduced incrementally to produce different levels of bit errors experienced at the receiver of the graphics card. The relationship of BER level vs. transmit swing was first established by a loopback testing mode on the system similar to the one described above.

Then, using the same transmit swing settings, the performance of each of the three benchmarks was measured. Using this approach, performance vs. BER experienced at the graphics card receiver could be characterized.

TABLE II. MINIMUM BER LEVELS AT GRAPHICS RX THAT INDUCED 3% AND 50% PERFORMANCE LOSS ON WORST-CASE BENCHMARK

Graphics Card	**BER for Graphics Rx performance loss**	
	3%	***50%***
A	$1x10^{-8}$	$1x10^{-6}$
B	$3x10^{-6}$	$6x10^{-5}$
C	N/A	N/A
D	N/A	N/A

Table II summarizes the performance loss observed as a function of BER at the Graphics Rx. Table III lists similar information as a function of BER at the CPU Rx. Values are reported for both 3% performance loss and 50% performance loss. Note that cards C and D were extremely robust to low CPU Tx voltage swing and did not encounter bit errors even at the lowest swing settings. Therefore it was impossible to characterize performance vs. Graphics AIC Rx BER on cards C and D using this technique.

TABLE III. MINIMUM BER LEVELS AT CPU RX THAT INDUCED 3% AND 50% PERFORMANCE LOSS ON WORST-CASE BENCHMARK

Graphics Card	**BER for CPU Rx performance loss**	
	3%	***50%***
A	$1x10^{-7}$	$3x10^{-6}$
B	$1x10^{-4}$	$2x10^{-4}$
C	$5x10^{-7}$	$2x10^{-6}$
D	$4x10^{-8}$	$3x10^{-7}$

The first notable point is that even a 3% performance loss was not observed until a BER of at least 10^{-8}, which is four orders of magnitude above the PCIe BER spec of 10^{-12}.

Second, although the table values represent the worst-case benchmark, there was not a large difference in the behavior of different benchmarks in terms of relative performance loss. This is shown in Fig. 2, which depicts the BER vs. performance loss at the CPU receiver when using PCIe card D. Also evident in Fig. 2 is the typical number of BER sample points and intervals that produced the data summarized in Tables II and III. This card showed the most difference between benchmarks, but as can be seen, even on this card, the relative performance loss is roughly equivalent across all three benchmarks at a given BER.

Another observation is that once performance starts to degrade on the order of 3%, it does not require a much greater BER to degrade the performance significantly further. This can be seen in Tables II and III, or graphically in Fig. 2. The latter graph illustrates that each benchmark performance metric degrades by 50% at a BER only 1 to 1.5 orders of magnitude above the 3% degradation point.

However, there were some differences in CPU Rx BER and Graphics Rx BER in this regard. Fig. 3 depicts this finding for Card A running Unigine Heaven. Graphics Rx BER starts to produce performance problems at a level roughly two orders of magnitude below CPU Rx BER, but the performance decrease after that point is more gradual than CPU Rx, such that at BER levels in the vicinity of 10^{-6}, performance penalties are similar.

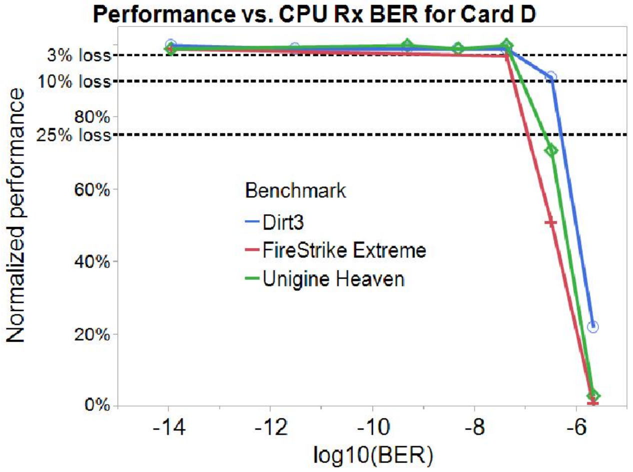

Figure 2. PCIe Card D performance vs. BER on each of three benchmarks

It should also be mentioned that some card-to-card differences were observed. This is shown in Fig. 4, which separately delineates the Unigine performance vs. CPU Rx BER for each card. Although card B had the lowest performance, it proved to be the least affected by bit errors, with little degradation all the way up to 10^{-4} BER. It could be speculated that the lower performance of this card resulted in lower utilization of the maximum available bandwidth on the PCIe link, thus preserving some additional bandwidth to compensate for the error retries on the link. However, this would not explain why card A, the highest performing card, showed the second-most resilience to bit errors in terms of performance impact. This suggests there are other factors that create these differences from card to card.

B. PCIE Gen 3 link between CPU and test add-in card

Another round of experiments was designed with a PCIe Gen 3 test add-in card to understand the BER levels associated with serious performance degradation. Voltage sampling and timing sampling points on the CPU PCIe receiver were offset from nominal values to induce a bit error ratio in the digital loopback mode described in the previous section, in order to establish the relationship of BER to the margin offsets.

Next, a PCIe functional test mode was utilized, in which the CPU wrote pre-defined data to the add-in card with all sampling points at nominally trained values. While reading back the data from the card, the CPU receiver margin hooks were operating to test at different sampling offset points, and error reporting was enabled to give visibility into detected receiver errors such as bad packets and CRC errors. In addition, data mismatch errors escaping the PCIe error detection mechanisms were identified by comparing the received bits against the transmitted bits in the CPU memory. In this way, the BER level creating normally undetected data mismatch errors could be empirically measured. The experiment was performed once with the timing sampling offset used to induce BER, and again with the voltage sampling offset used to induce BER at the CPU receiver.

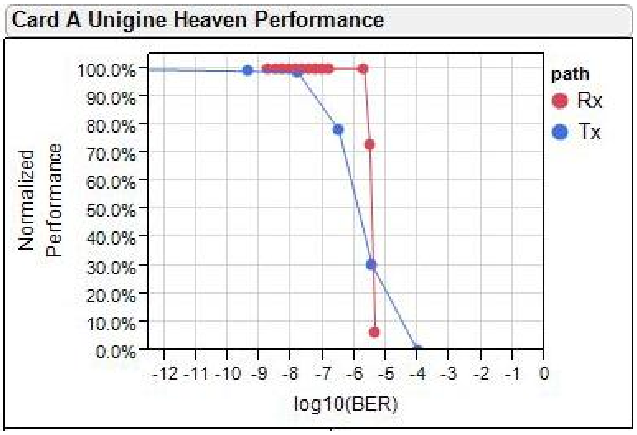

Figure 3. Unigine performance vs. BER on either CPU Rx or Tx vs. BER on PCIe graphics Card A

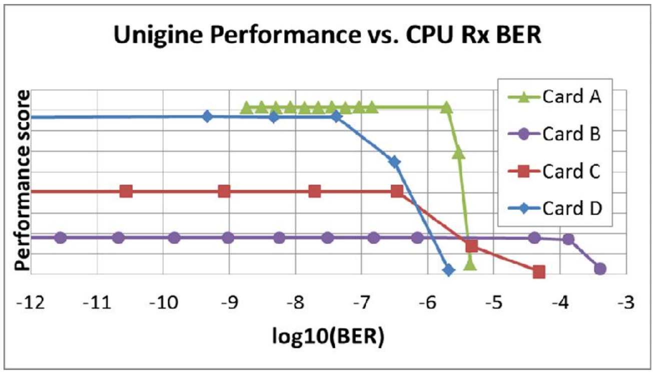

Figure 4. Unigine performance vs. CPU Rx BER on each of four PCIe graphics cards

Table IV shows the comparison of BER levels resulting in data mismatch errors during the PCIe functional test versus the BER levels causing a 100% performance loss. It should be mentioned that with the PCIe functional test content running in this part of the experiment, the 100% performance loss in actuality resulted in a crash or link hang requiring a reboot.

When BER was induced by changing the timing sampling point, the resolution was not sufficient to distinguish any data mismatch errors before reaching a BER that caused 100% performance loss. When using the voltage sampling offset, on 8 of the 120 runs, data mismatch errors were distinguishable before a crash occurred. On the other 112 runs, the high level of BER created a crash before any mismatch problems occurred.

Evident from these results is that any data mismatch issues escaping the built-in error detection mechanisms on PCIe Gen 3 occur at a BER very close to or higher than the BER that causes catastrophic performance problems. This is supporting evidence that as BER is increased, the main area of concern for an end user is in fact performance degradation rather than undetected data mismatch issues.

C. PCIe Gen 3 link between CPU and PCH

In this experiment, a 2nd generation Intel Xeon E5 processor was connected to an Intel BD82C606 Server Chipset Platform Control Hub (PCH) via a PCIe Gen 3 uplink. The intent was to study the impact of PCH Rx BER on performance of the uplink.

TABLE IV. AVERAGE BER AT WHICH 100% PERFORMANCE LOSS OR DATA MISMATCH ERRORS OCCURRED ON PCIE GEN3 ADD-IN CARD

Method used to induce BER	Avg BER for 100% performance loss	Avg BER for data mismatch error
Timing sampling offset	3.0×10^{-7} (120 runs)	Not measurable
Voltage sampling offset	4.8×10^{-7} (120 runs)	5.6×10^{-7} (measurable on 8 of 120 runs)

First, in order to monitor the performance, a benchmark test was run that was known to exercise the bandwidth of the PCIe link. While this was done, jitter of various amplitudes was injected at the receiver to induce a BER at the PCH Rx. While the jitter was injected, error logs were utilized to monitor the rate of CRC and link recovery errors with respect to the total number of bits transmitted to calculate the effective BER at that jitter amplitude setting. It was found that the jitter injection provided only a coarse control over the effective BER. Finer granularity was achieved by complementing the jitter injection with voltage and temperature adjustments, which provided a finer adjustment to the receiver BER level. This way, performance penalty vs. PCH PCIe uplink receiver BER could be characterized.

The jitter injection technique plus voltage & temperature adjustment did not provide as fine of control over the BER as the sampling point adjustment technique used in part A. However, this technique did have the advantage of being able to monitor the actual bit errors occurring during the performance test runs themselves.

Fig. 5 displays the results of this experiment. The region of >3% performance penalty was witnessed to be in the vicinity of 10^{-10} BER, again implying there is some buffer between a performance issue and the 10^{-12} BER specification. Because of lack of precise control over the BER with the jitter injection, there was a clear absence of data points in the BER range of 10^{-9} and 10^{-4}. Somewhere in this range, and by the time 3×10^{-4} BER is reached, the part is not able to function, which is represented by the 100% performance penalty on the graph in Fig. 5. Because of the sparseness of the data points, it is not known at exactly what BER this occurs. Based on the slope of the points at or around ~10^{-10}, it appears that 50% degradation would occur in the low 10^{-9} range. This agrees with the PCIe graphics card experiment in section A, which also showed a performance degradation from 3% to 50% occurring within approximately 1-1.5 orders of magnitude change in Rx BER.

D. SATA 6Gb/s link between PCH and hard disk drive

For this measurement, an Intel BD82C606 Server PCH SATA 6 Gb/s link attached to a hard drive was studied. Similar to the experiment in the previous section, jitter injection was used at the PCH Rx to induce a BER. Jitter frequency and amplitude changes were made to vary the BER, and for finer adjustments, temperature adjustments were made in addition to a validation test hook that provided some level of control of the PCH Rx voltage sampling point with respect to the center of the data eye. As these adjustments were being made, the BER could be calculated by logging the disparity and CRC errors occurring on the SATA link and dividing by the total number of bits transmitted.

While errors were being induced in this manner, a performance benchmark involving continuous reads and writes to the hard drive was utilized to stress the SATA I/O as well as monitor the performance at various levels of BER. With the combination of jitter injection and the data eye margining hook, a reasonable level of accuracy was achieved in inducing different levels of BER on the SATA link. Similar to the PCH PCIe uplink experiment, errors were induced and monitored while the performance monitor itself was being run.

Fig. 6 shows the outcome of the experimental measurements. As BER was increased above the spec of 10^{-12}, minimal overall performance degradation was witnessed until a BER level of approximately $3x10^{-10}$ was achieved. At this level, a 3% performance penalty was observed, but from that point on, the rate of performance degradation with respect to BER increased dramatically. As was so often witnessed in the experiments reported in this whitepaper, 50% performance degradation occurred only at a BER level one order of magnitude higher, at approximately $3x10^{-9}$.

IV. IMPLICATIONS AND FUTURE WORK

The data presented here suggests for the high speed serial bus types studied, there are at least two orders of magnitude of margin above the max BER specification before a user would experience any noticable performance loss from replaying data after an error is detected. It is worth mentioning, however, that the empirical data sometimes showed lower margin than a simple latency-based theoretical projection would predict. Murali et al. [7] speculate that based on error retry-related latency penalties, average observed latency would not show degradation until a packet or flit error ratio in the range of 0.1%-1%. In this context, latency refers to the amount of additional delay in the data packet, or “flit,” that is created by the receiving device notifying the transmitting device of the CRC error as well as the resend of the correct data by the transmitting device. Projecting this value onto PCIe Gen 3, for example, with a typical CRC-protected packet payload size of 1200-2200 bits for the products measured in this paper, one would predict there would be no performance concern until a BER elevates to the range of $\sim 10^{-7}$ to 10^{-6}. Yet in some of the experiments, performance began to show measurable decrease in the neighborhood of 10^{-8} or even 10^{-10}.

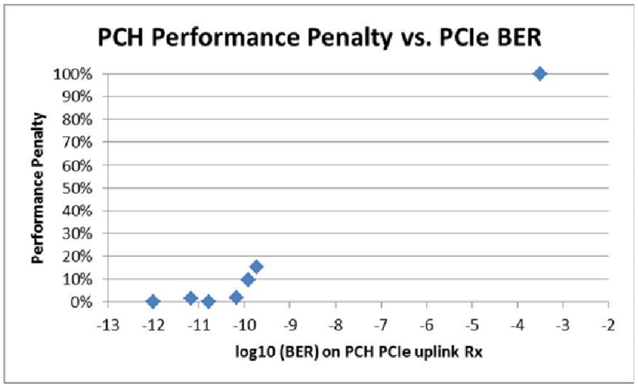

Figure 5. PCH performance penalty vs. PCH Rx BER on the PCIe uplink to the CPU

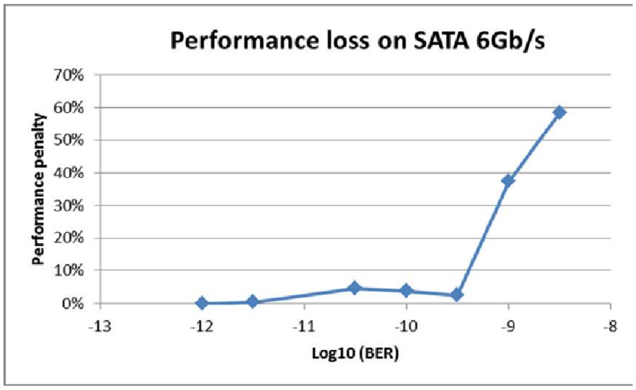

Figure 6. Performance loss vs. PCH Rx BER on the SATA 6Gb/s link to a hard drive

This suggests that true effective latencies with real modern-day products and workloads, taking into account the error profiles (for example, number of consecutive packets with errors), are sometimes greater than the assumed penalties in [7]. While error ratio profiles could differ by scenario and would not always match those in the reported experiments, the fundamental sources of bit errors in the experiments (jitter, elevated temperature, reduced transmit voltage swing, and a non-centered data sampling point) are all sources that could be experienced in a real-world system.

To minimize the impact of extended test runs and using more expensive design solutions to ensure parts meet the BER spec, an alternative approach to a simple spec value would be to architect in the right validation hooks and capabilities to measure performance changes as data eye margins decrease or alternatively, as BER increases. Validation activities can then concentrate on checking that the vast majority of parts and systems will not experience noticeable performance penalties—for example, no more than 3% performance loss—from resending data across the link as a result of error detection. When needed, test content such as the PCIe Gen 3 functional test used in this paper can be used to confirm that undetected data mismatch errors happen at or above BER levels that create severe performance degradation or hangs.

By structuring validation targets with respect to performance, product validation teams can have confidence they are truly validating for a quality end user experience, rather than a generic BER level. To illustrate, the BER requirement of 10^{-12} is prevalent in specs for high speed serial buses, despite different levels of error detection and different retry time penalties on these various buses, not to mention product-level architectural differences that could create different retry penalties product to product on the same serial bus type. By forcing products to abide to one generic BER spec that is not explicitly tied to an end user impact, the spec level must be overly conservative to account for all possible factors across all possible systems, implying that most products are over-designing and over-validating.

TABLE V. TEST TIME DIFFERENCES AT DIFFERENT BER LEVELS

BER requirement	Min test time for 95% confidence, PCIe G3 (seconds)	Min test time for 95% confidence, SATA 6 Gb/s (seconds)
10^{-12}	374	499
10^{-10}	3.74	4.99

In contrast, by aligning to a performance-based requirement, this conservatism can be avoided, resulting in additional design margin and shorter validation time. Design margin benefit is extremely difficult to quantify even on a single I/O type because of the enormous variety of Si circuit designs, fabrication processes, and board designs. Validation time benefit is more straightforward to quantify, however. To empirically confirm that a given link is less than or equal to a certain BER at a certain confidence level, one must test for a sufficient time. The Poisson probability distribution may be used to calculate the required length of test time to validate against a certain BER to a level of 95% confidence, assuming no errors are encountered during the test:

$$Min\_Test\_Time = \frac{-\ln(1-0.95)}{BER \times dataRate}. \quad (2)$$

Table V shows the test time improvement for PCIe Gen 3 and SATA 6 Gb/s using this approach. If an empirical validation test of this nature was implemented in a manufacturing test, for example, this would imply a 99% reduction in test time if it were confirmed that only a BER of 10^{-10} was needed as opposed to 10^{-12}. This is immediately evident from (2): test time is inversely proportional to BER.

One challenge encountered in this study was that, as far as the authors were able to discern, there is no published experimental data of performance penalties vs. BER on modern high-speed interconnects to which a comparison could be made. All previous investigations on this subject appear to be purely theoretical ([7][9]) and did not even analyze a specific existing high speed interconnect type. Because this appears to be an area not previously explored, future studies will include other high speed interconnect types besides PCIe and SATA, as well as other scenarios for PCIe that include a graphics card AIC where hardware acceleration is turned off. It is also the hope that this work will motivate others in the industry to perform studies on their platform architectures.

V. SUMMARY

In this paper, four experiments were conducted to study the impact of increasing levels of BER on performance of high speed serial buses. On a PCIe Gen 3 link running between a CPU and a graphics add-in card, it was found that although there were some card-to-card differences, performance did not start to decrease from error-induced retries until a BER of 10^{-8} at the lowest. On a PCH PCIe Gen 3 uplink to a CPU as well as a SATA 6 Gb/s I/O running from a PCH to a hard disk, performance did not appreciably decline until a BER of 10^{-10} or higher. Finally, the PCIe Gen 3 functional test between a CPU and test add-in card showed that catastrophic performance issues arose at a BER of ~10^{-7} but that undetected mismatch errors do not occur until the same level of BER or worse.

The data suggests that many products have additional margin above the 10^{-12} BER spec before any user impact would occur. If new standards and practices were adopted to validate against performance impact instead of a generic BER specification level, conservatism leading to costly over-design and over-validation could be avoided.

ACKNOWLEDGMENT

The authors would like to thank Alejandro Cardenas, Federico Hernandez Reyes, David Steele, and Dale Robbins for performance measurements on the PCH PCIe uplink and SATA. We would also like to thank Tsafrir Waller for the CPU/AIC PCIe performance test runs and analysis.

REFERENCES

[1] PCI Express® Base 3.0 specification, www.pcisig.com [retrieved: Aug, 2014].

[2] IEEE 802.3TM-2012 Section 5, standards.ieee.org [retrieved: Aug, 2014].

[3] Serial ATA Revision 3.1 specification, www.sata-io.org [retrieved: Aug, 2014].

[4] Universal Serial Bus 3.1 Specification, www.usb.org/developers/docs [retrieved: Aug, 2014].

[5] W. W. Peterson and D. T. Brown, "Cyclic codes for error detection," Proc. IRE, vol. 49, Jan. 1961, pp. 228-235.

[6] G. Castagnoli, S. Bräuer, and M. Herrmann,"Optimization of cyclic redundancy-check codes with 24 and 32 parity bits," IEEE Transactions on Communications, vol. 41, June 1993, pp. 883-892.

[7] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. De Micheli, "Analysis of error recovery schemes for networks on chips," IEEE Design and Test of Computers, Sep-Oct 2005, pp. 435-442.

[8] W. Liu and W. Lin, "Additive white gaussian noise level estimation in SVD domain for images," IEEE Transactions on Image Processing, vol. 22, pp 872-88.

[9] S. Wang, S. Sheu, H. Lee, T. O, "CPR: A CRC-Based Packet Recovery Mechanism for Wireless Networks," 2013 IEEE Wireless Communications and Networking Conference, April 2013, pp. 321-326.