DesignCon 2011

AMI Simulation with Error Correction to Enhance BER

Xiaoqing Dong, Huawei Technologies
Dongxiaoqing82@huawei.com

Geoffrey Zhang, Huawei Technologies
geoff.zhang@huawei.com

Kumar Keshavan, Sigrity Inc.
ckumar@sigrity.com

Ken Willis, Sigrity Inc.
kwillis@sigrity.com

Zhangmin Zhong, Sigrity Inc.
zmzhong@sigrity.com

Adge Hawes, IBM
ADGE@uk.ibm.com

Abstract

The ultimate goal for a high speed serial link is to lower the system Bit Error Rate (BER). While hardware results have the final say, simulation plays a key role in checking the system feasibility and predicting the performance margin. Systems applications often use SerDes components from multiple suppliers, requiring IBIS-AMI models for simulation interoperability. AMI simulation can be used to identify the marginal serial links in the overall system, among which are those that do not meet BER requirements. The BER of these links can be improved by the utilization of forward error correction (FEC) algorithms. This paper will present an end-to-end methodology in which AMI modeling techniques and existing serial link analysis is augmented with FEC to improve BER performance.

Author(s) Biography

Xiaoqing Dong joined Huawei Technologies in 2006 as a signal integrity research engineer, where she works on high speed active SI simulation and measurement technology. She received her bachelor and master degrees in communications and information system from Harbin Institute of Technology, for research in Information and Communication Engineering.

Geoffrey Zhang has been with Huawei Technologies since 2009. He is currently the CTO for the Interconnect Division. Prior to joining Huawei he worked for Texas Instruments, Lucent Technologies, Agere Systems, and LSI Corporation. He has worked on various products including data converters, timing devices, read channel chips, and SerDes cores. He received his Bachelor and Master degrees both in the Department of Electrical Engineering from Zhejiang University, China, and Ph.D. in microwave engineering from Iowa State University.

Dr. Kumar Keshavan is Software Architect at Sigrity. Dr. Keshavan's expertise includes circuit simulation, circuit extraction, channel simulation, and timing analysis. Dr. Keshavan has 25 years in the EDA industry. In the past he has worked as software architect at Cadence Design Systems. He was one of the founders of Velio, Inc. which was a pioneer in providing SerDes switching devices to the communication industry. Dr. Keshavan has also been an active participant in the EIA IBIS standards organization. More recently he was one of the initiators of the IBIS AMI standards for serial links.

Ken Willis is a Product Marketing Manager at Sigrity, responsible for advanced signal integrity solutions. He has more than 20 years of experience in the modeling, analysis, design, and fabrication of high-speed digital circuits. Prior to Sigrity, Ken held engineering, marketing, and management positions with Tyco, Compaq Computers, Sirocco Systems, Sycamore Networks, and Cadence Design Systems.

Zhangmin Zhong joined Sigrity as a Senior Application Engineer in 2009. Prior to joining Sigrity he worked for IO Methodology Inc., Cadence Design Systems, Alcatel-Sbell and Huawei Technologies. He has worked on various products and focused on high speed circuit designs. He received his Bachelor and Master degrees both in the Department of Electrical Engineering from Sichuan University, China.

Adge Hawes is a Development Architect for IBM at its Hursley Labs, United Kingdom. He has worked for IBM for more than 30 years across such hardware as Graphic Displays, Printing Subsystems, PC development, Data Compression, and High-Speed Serial Links. He has represented the company in many standards bodies such as PCI, SSA and Fibre Channel. Currently he develops simulators for IBM's High Speed Serial Links.

Introduction

With the release of the IBIS 5.0 specification, Algorithmic Modeling Interface (AMI) syntax has provided an industry standard means with which to model the complex adaptive equalization functionality seen in modern SerDes (serializer/deserializer) devices. For example, transmitter Feed Forward Equalization (FFE, or pre-emphasis) and receiver Decision Feedback Equalization (DFE) can be efficiently modeled with AMI techniques. This has provided simulation interoperability for engineers working with SerDes from multiple suppliers.

Few serial links are designed completely from scratch; most have restrictions due to cost or compatibility (ex. existing PCBs, backplanes, connectors) that often limit significant BER improvements. Limited by legacy hardware, electrical engineers are still able to improve system performance by certain “soft” methods implemented in ASIC front-end logic or on-board Field Programmable Gate Arrays (FPGA), such as line coding and error correction. Used extensively for optical links, error correction methods are now beginning to find application in multi-gigabit electrical serial interfaces.

A prototype flow for the incorporation of error correction into serial link analysis is proposed and exercised in a case study. The proposed flow is as follows:

1. For the channels of interest, S-parameter models are acquired from measurement using a high bandwidth Vector Network Analyzer (VNA)
2. To produce raw bathtub curves (without error propagation), serial link signal integrity analysis is performed using commercial EDA (electronic design automation) software and IBIS-AMI models
3. Marginal or failing serial links are identified as candidates for FEC
4. Performing error propagation calculation; during this process the converged DFE tap coefficients from simulation are used to compute the burst error voltage offsets, which in turn serves as the basis for FEC performance analysis, producing the burst error order of the serial link
5. Error correction analysis, to predict the BER performance enhancement

In multi-gigabit serial links, DFE is often used at the SerDes receiver for reducing channel inter-symbol interference (ISI) and enhancing link performance. However, DFE can also introduce problems such as error propagation, which can lead to burst errors. The DFE operation can provide some key insight into the burst errors that contribute significantly to BER degradation. A “slicer” algorithm can be utilized together with the DFE coefficients to calculate error propagation, and predict the burst error order of the serial link.

With the cumulative BER performance computed, including burst errors, forward error correction (FEC) algorithms can be applied to examine potential improvements in BER. These FEC algorithms can include:

- FEC that deals with random errors (BCH code)
- FEC that deals with single burst errors (Fire code)
- FEC that deals with multiple burst errors (RS code)

Both simulation and experiments to date have shown that particularly bad channels will fail their associated BER requirements even with FEC on. However, other marginal channels have shown an improvement in BER margin by as much as 10^3. These initial trials indicate FEC may not be a panacea, but may be an effective way to enhance the BER margin for marginal channels in existing hardware.

This paper will cover the entire end-to-end serial link simulation process, which includes:

- Current “standard” serial link simulation process
- IBIS-AMI modeling, and the utilization of these models in simulation
- Bathtub curve generation for initial BER prediction by EDA tools
- Eye quality metric – weighted eye
- Error correction theory and methods
- Prediction of BER improvement expected by utilizing error correction techniques

Serial Link Simulation Process

The current state-of-the-art for multi-gigabit serial link signal integrity (SI) simulation involves convolution of the channel's impulse response with a large bit stream stimulus to produce time domain waveforms. The "channel" in this context refers to the analog circuit comprised of the output stage of the SerDes transmitter (Tx), the input stage of the SerDes receiver (Rx), and the passive interconnect between the two. This interconnect can consist of circuits for the printed circuit board (PCB), packages, connectors, etc. as shown in the Figure below.

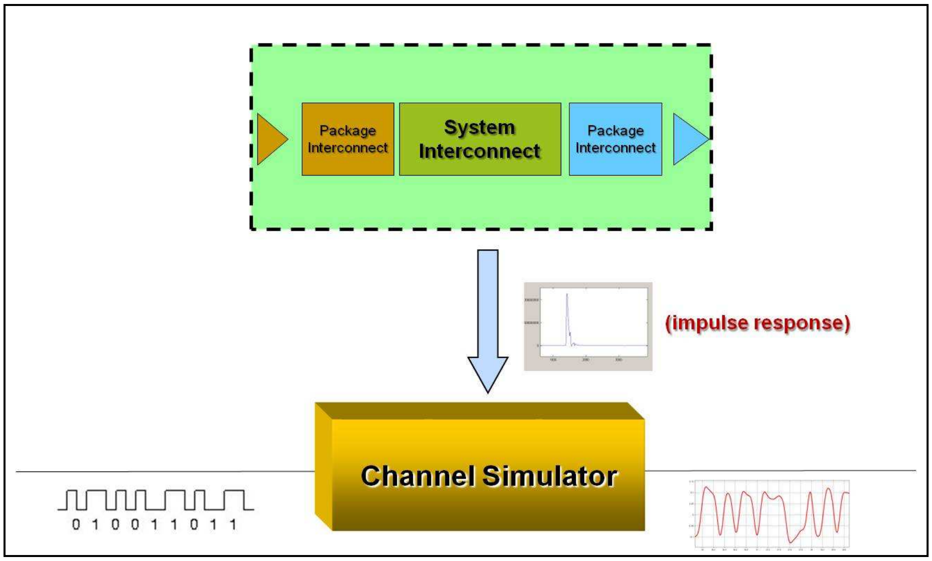

Figure 1 – Serial link simulation flow

The impulse response can be generated by either frequency or time domain techniques. The channel can be represented with S-parameters, and inverse FFT (Fast Fourier Transform) techniques can be used to produce the impulse response. Also, traditional time domain Spice circuit simulation can be used to produce a step response, from which the impulse response can be directly computed. Using advanced convolution techniques, simulation can be both very fast and very accurate, enabling the simulation of a million bits or more in a minute or two.

IBIS-AMI Modeling

With the release of the IBIS 5.0 standard, algorithmic modeling for SerDes equalization became a standard practice, allowing the adaptive equalization behavior of SerDes devices to be efficiently included in SI simulations. There are essentially two APIs (application programming interface) defined in the IBIS standard as part of the Algorithmic Model Interface (AMI):

- AMI_Init > allows impulse response modification
- AMI_GetWave > allows waveform modification

These are shown in the figure below.

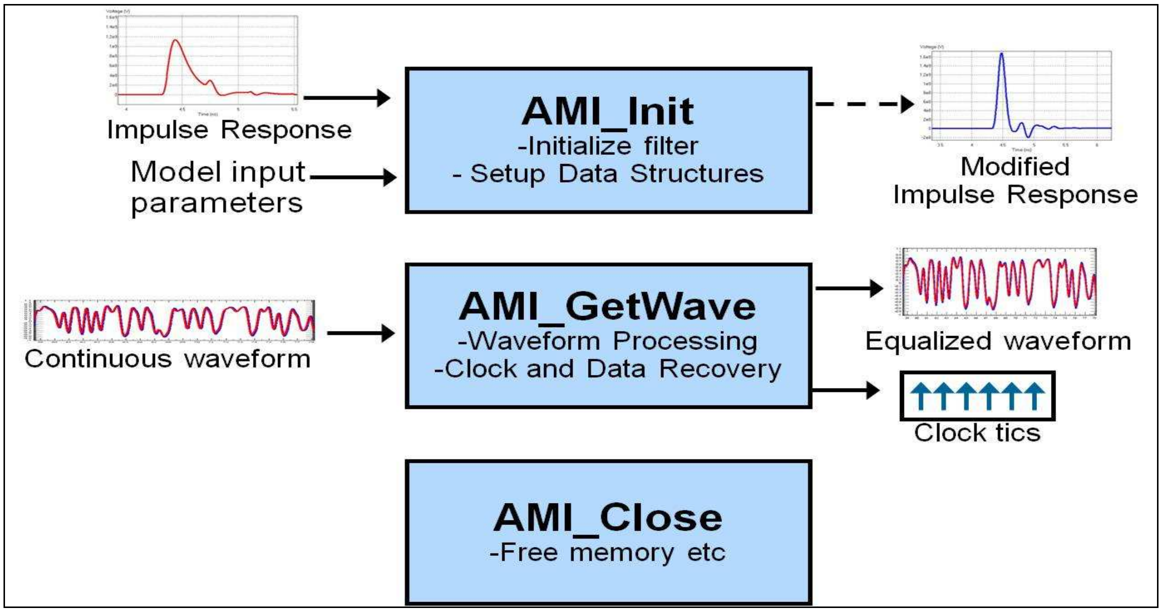

Figure 2 – IBIS-AMI APIs

The AMI_Init function is used for equalizers (EQ) that perform one-time adaptation, meaning that they adapt their EQ settings once for the specific channel. This is common practice for SerDes transmitters that use Feed Forward Equalization (FFE, pre-emphasis, de-emphasis). The AMI_GetWave function is used for real-time adaptive EQs, and finds common application in receiver Decision Feedback Equalization (DFE), where the EQ is constantly adapting based on the waveforms it receives.

So how would this fit in with the serial link simulation methods described in the previous section? Let's take for example a case where the Tx uses an algorithmic model with FFE and the Rx uses one with DFE. The sequence of events for simulation would be as follows:

- Impulse response generation (with EQs off)
- Impulse response is modified by the Tx algorithmic model (AMI_Init function)
- Modified impulse response is convolved with the bit stream to produce waveforms at the Rx (channel simulation)
- Waveforms are passed to the Rx algorithmic model
- Rx algorithmic model applies equalization, and passes back modified waveforms and sampling data

These modified waveforms are then post-processed to produce eye diagrams and other outputs. This is shown in the figure below.

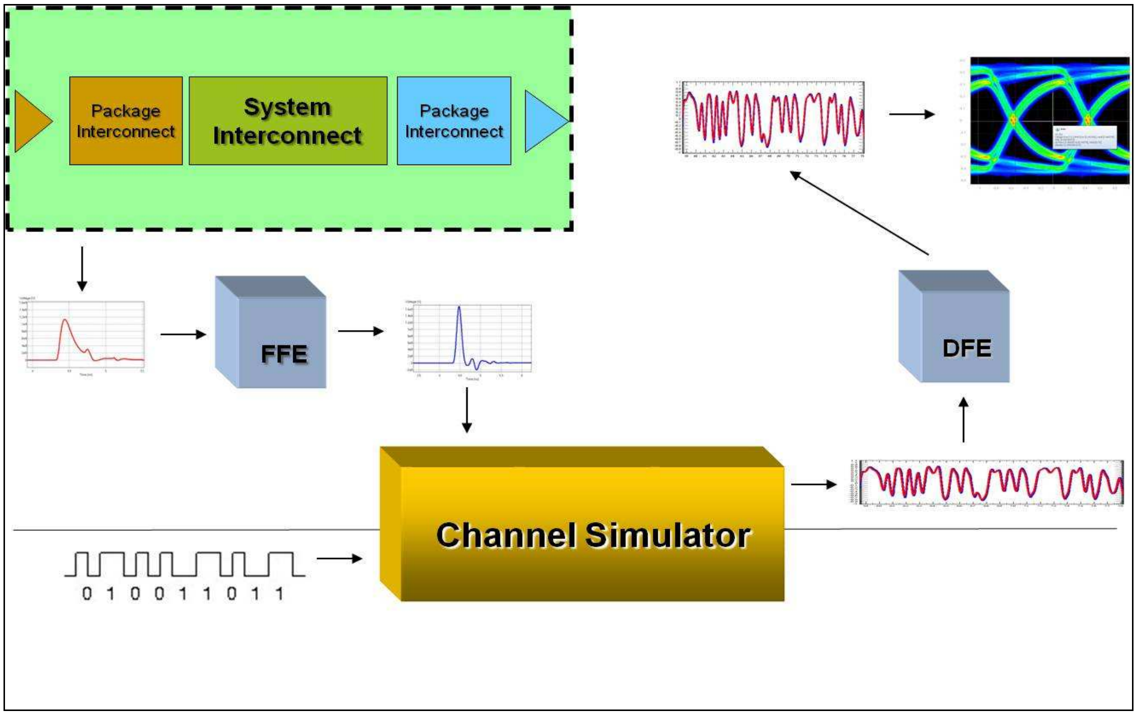

Figure 3 – AMI models in simulation flow

In practice, this flow works well, and can faithfully reproduce transistor-level waveforms, as shown in the figure below. This shows a comparison between transistor-level Spice simulation and channel simulation, using a behavioral Spice circuit model together with an IBIS-AMI algorithmic model.

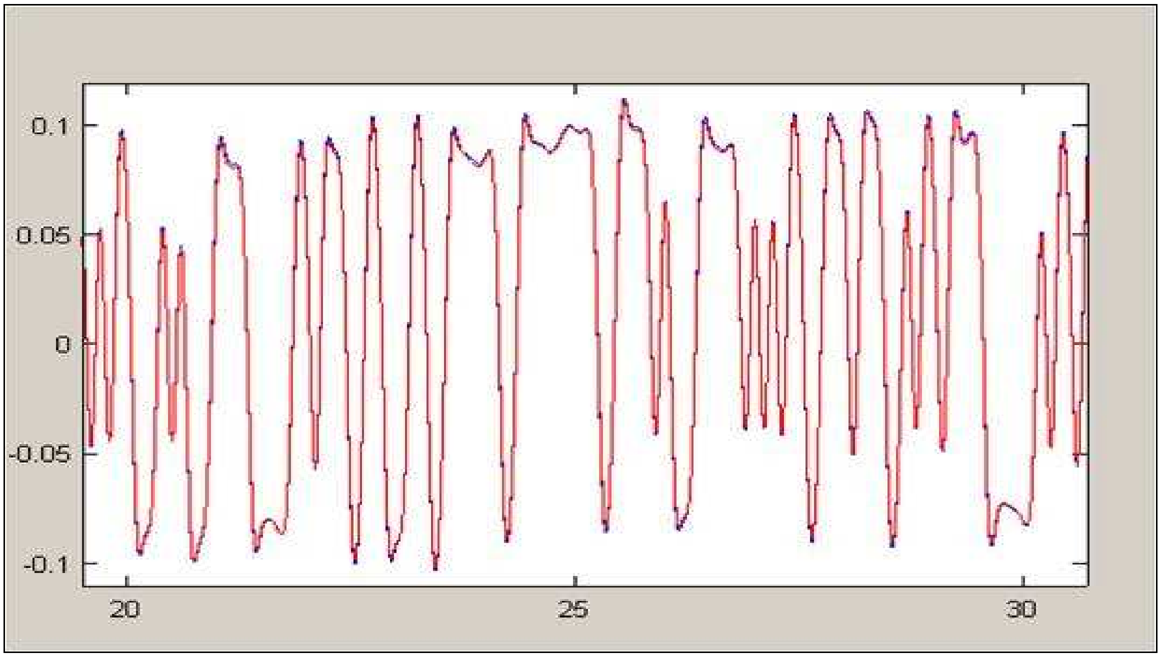

Figure 4 – Transistor-level Spice simulation vs. channel simulation

Bathtub Curve Generation

A key objective of serial link simulation is to assess the bit error rate (BER) [1,2], given a sampling eye mask, or conversely to find out the eye width, eye height, or even eye area as a function of BER.

The first step in assessing BER is to obtain the “eye density” data at the receiver sample point. An example of this is shown below.

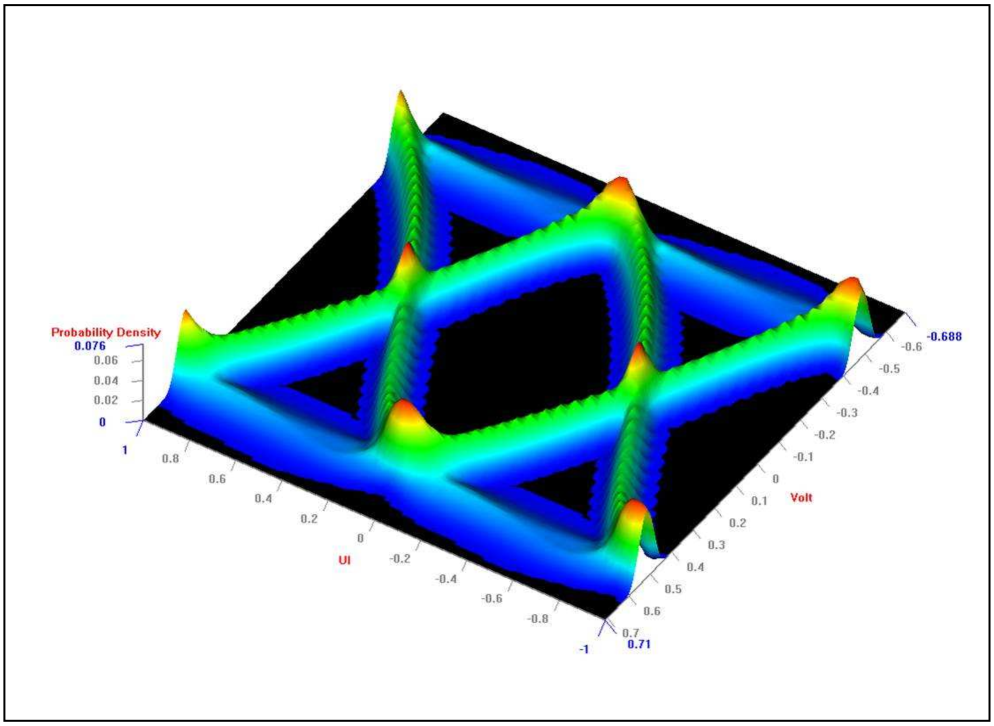

Figure 5 – Eye density plot

Eye density is obtained through serial link channel simulation and will include effects of the channel components, Tx and Rx equalization (using IBIS AMI models), crosstalk effects, and addition of purely random jitter and noise. In principle, BER is simply the ratio of the number of symbols inside the eye mask to the total number of symbols inside one unit interval (UI) width. In practice however, we are interested in low numbers for BER, often < 10^-12, which involves a rather large dynamic range. Most popular BER algorithms are computed near the center of the eye, yielding the well known “bathtub curve”, which gives the relationship between BER and the eye width [1,2].

A bathtub curve is actually an extrapolated cumulative density at the center slice of the eye. The extrapolation is based on the assumption that the tail of the distribution is Gaussian in nature. One of the best known mathematical approaches to derive BER is the Dual Dirac method. Based on Gaussian tail extrapolation, the bathtub curve can be characterized by its intersection and slope. The intersection is proportional to deterministic jitter, whereas the slope represents random jitter. Figure 6 shows an example of a bathtub curve [1].

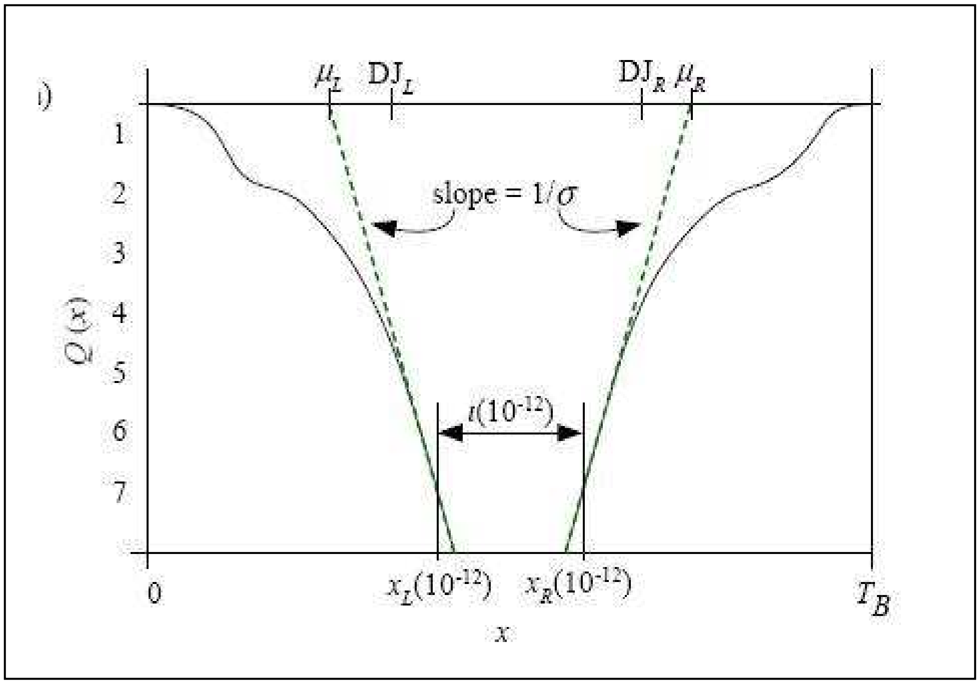

Figure 6 – Bathtub curve

Eye Quality Metric – Weighted Eye

The primary objective of the time domain channel simulation is the generation of the eye distribution, as shown in the figure below. Once the eye distribution is obtained it can be analyzed using various metrics. These metrics can quantify the quality of the serial link simulation results, and provide a means by which to compare multiple scenarios. These metrics can range from simple eye opening (eye contour) to bit error rate measurements in the voltage and time domains. The basic idea is to quantify the "sparseness" of the eye distribution. Thus the sparser the eye, the better the eye opening and the lower the BER.

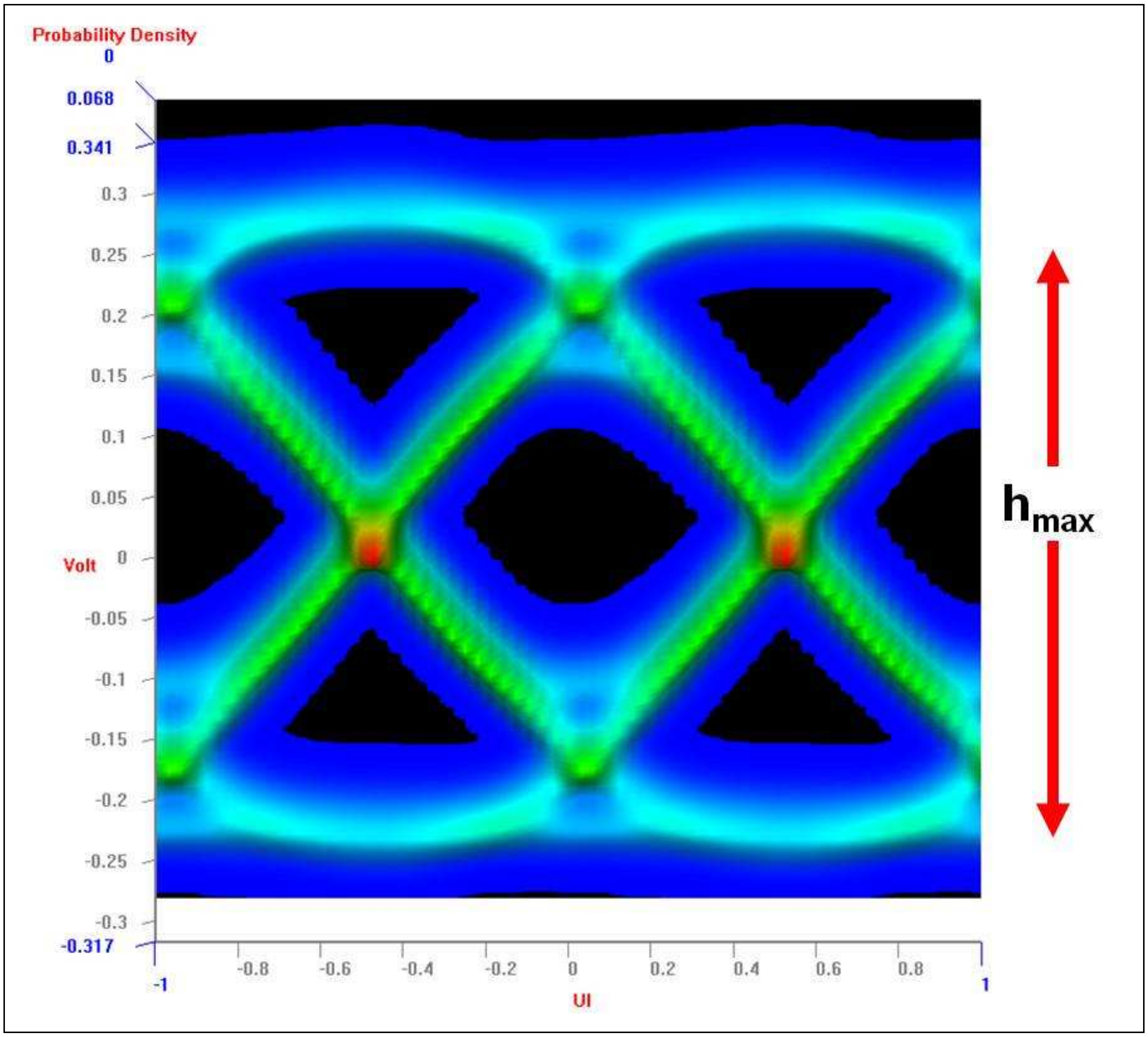

Figure 7 – Eye distribution

The eye quality metric used in this study is called “weighted eye opening”. The weighted eye opening is measured at the sampling point (usually at the center of the eye) and is given by:

$$\text{Weighted\_eye} = \sum h(y)\, p(y)$$

Where h(y) is eye height, and p(y) is the probability. The maximum possible eye opening h_{max} , which is the outer extent of the eye measurement, can be used to normalize the weighted eye. A larger weighted eye is always superior.

The advantage in using this metric lies in its simplicity, and the fact that it will always have a non-zero value. In addition, when normalized as is the case here, the weighted eye is also an excellent means to quantify the effect of equalization, as well as the effect of the various components that comprise the channel.

Error Correction Theory and Methods

Traditionally, link errors through backplane channels have been considered to be dominated by random errors, mainly due to random jitter and random noise. However, as the data rates increase beyond today's mainstream 3Gbps to 6Gbps ranges, complex signal processing techniques are increasingly used, such as Decision Feedback Equalization (DFE) and line coding. As a result of the application of these techniques, link errors turn out to be more related to each other, with a mixed error occurrence mechanism of independent errors and burst errors.

In multi-gigabit serial links, DFE is often used at the SerDes receiver for enhancing link performance by reducing channel post-cursor inter-symbol interference (ISI). However, DFE can also introduce problems such as error propagation, which can lead to burst errors.

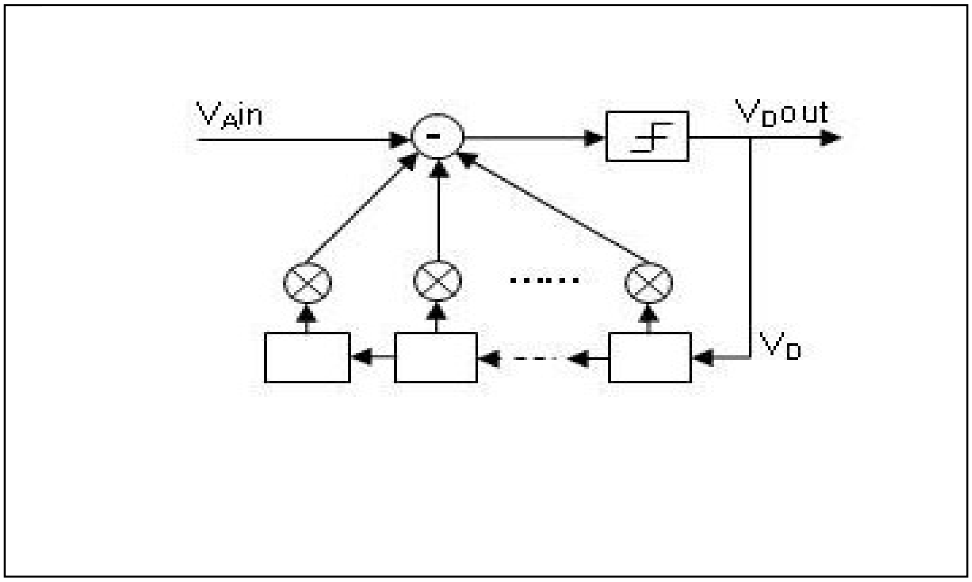

Figure 8 - Simplified assumption of feedback mechanism

As illustrated in Figure 8, the digital output of the feedback equalizer is:

$$V_D out(t_0) = V_A in(t_0) - DFE_1 \cdot V_D(t_{-1}) - DFE_2 \cdot V_D(t_{-2}) - ... - DFE_M \cdot V_D(t_{-M}) \qquad (1)$$

Where, $V_A in(t_0)$ is the input analog signal, $V_D(t_i)$ is the decision output of the ith bit, and DFE_i is the ith equalizer coefficient.

The DFE slicer output is directly related to the previously M detected bits. Once single or multiple errors occur in these previously detected M bits, they impact the current bit's detection with certain probability. This probability depends on the error number and error location in the previous M bits.

Serial links also utilize other techniques for reducing the risks associated with the worst patterns in line coding, such as data scrambling. The 10GBASE-KR standard specifies 2 stages of scrambling in the transmission direction; firstly the PRBS58 scrambling for 64b66b coding, and secondly the PN2112 scrambling for the transmission data blocks.

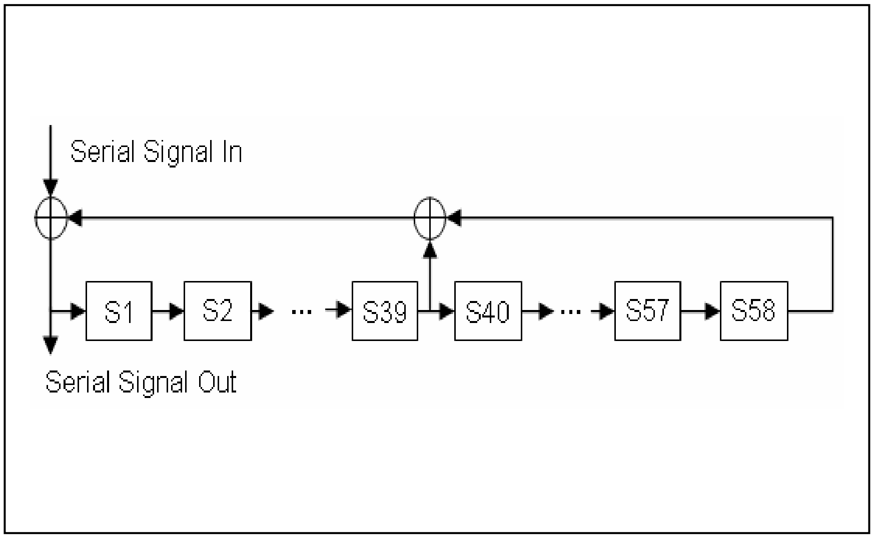

Figure 9 - Scrambler used in 10GBASE-KR

Although this scrambling optimizes the DC balance of the transmitted data, it potentially triggers error propagation once random errors happen in any of the scrambler registers.

With these complex sources of error now present in modern serial link interfaces, engineers are beginning to turn to forward error correction (FEC) techniques. Used extensively in optical links, this approach has begun to find a place in high speed electrical backplane applications as a means of improving serial link BER performance.

Error Propagation Calculation

From a simulation perspective, time domain processing is the most accurate way to account for the impacts of all different kinds of impairments in serial links. But even with state-of-the-art channel simulation, simulating 10^12 bits remains impractical. Statistical methods are therefore used to post-process the signal distribution calculated by channel simulation, to predict BER through bathtub curves. FEC calculations can be accomplished in a similar manner.

Inspired by the analysis model introduced in [3], which showed that error propagation can be modeled by probability calculation, the block error rate and bit error rate due to different error propagation lengths can be calculated by:

$$BER = \sum_{i=1}^{rll_{\max}} \sum_{allE} p(rll = i, E) \cdot W(E) \cdot p_1 \cdot (1 - p_1)^{n - rll_{\max} - i} \qquad (2)$$

where:

$rll_{\max}$ is the maximum error propagation length;

E is all the combinations of the error pattern when error propagation length is i

W is the probability that i bits in error among a n bit block

p_1 is the random error probability.

A critical aspect in estimating BER with error propagation is to calculate the probabilities of erroneous bits due to different propagation lengths, $p(rll = 1 : rll_{\max})$. These probabilities are gleaned from the raw voltage bathtub curves, by calculating the probabilities of error pattern E.

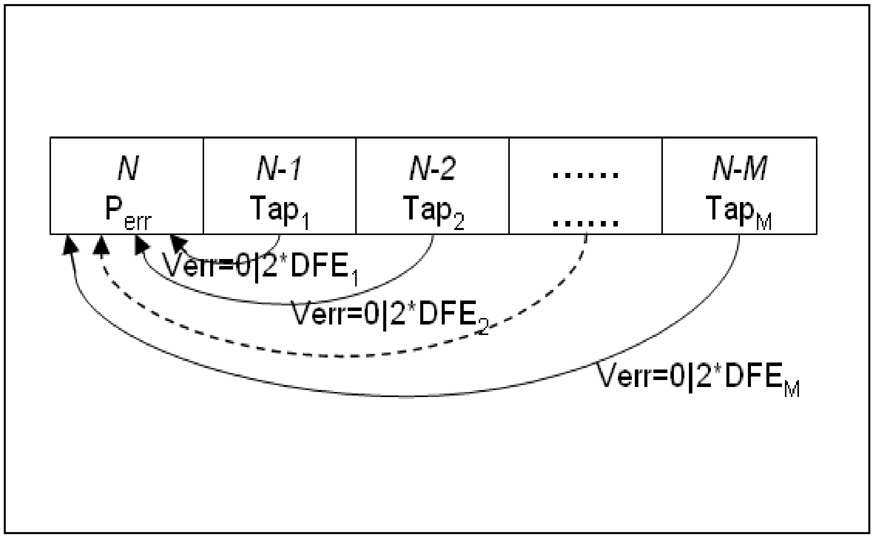

Figure 10 - Voltage offset induced by M tap DFE feedback loop (NRZ signaling)

From Figure 10, the occurrence probability of the voltage offset $2 \cdot DFE(i)$ induced by any of the DFE registers is therefore the bit error rate of that very register. There are 2^M total error patterns in a M tap DFE, where each pattern induces its own voltage offset to the current judging bit. It is obvious that the more DFE taps, the more error patterns, and the more complex the BER analysis becomes.

The voltage offset from a feedback loop is represented by:

$$[V_{sum\_err_j} \mid p_j] = \sum_{i=1}^{M} Verr(i) \qquad (3)$$

Where, $p_j = \prod_{j=1}^{2^M} [p_i \mid (1 - p_i)]$ when ith bit error using p_i else $(1 - p_i)$, and p_i is the bit error rate of the ith register of the DFE. When ith bit error, $Verr(i) = 2 \cdot DFE(i)$ else $Verr(i) = 0$.

The voltage offsets due to all the error patterns that exist in DFE feedback registers can be deduced using (3), and a map can be built to indicate the relationships between the error pattern E and the degraded BER using a noise bathtub at the slicer.

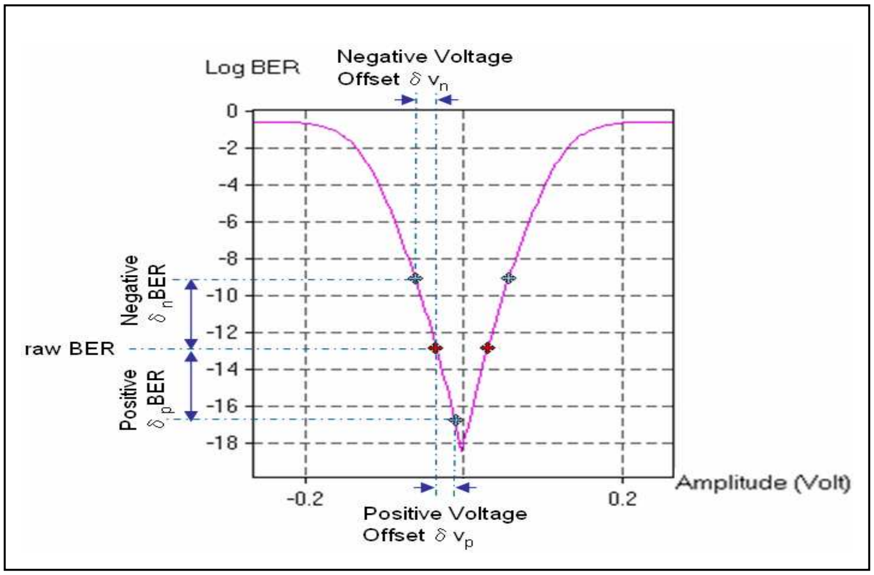

Figure 11 - Vertical bathtub curve and the voltage offset

Figure 11 gives the assumption that the diamond markers on the bathtub curve are located at the decision slicer levels, for example ±30mV. According to (3), a certain voltage offset δv can be estimated. Hence, BER due to this offset can be obtained directly from the bathtub curve. The BER due to error propagation should be the mean value of the BERs taken from the left and right hand side bathtub curves.

Notice that the voltage offsets can lead the current judging bit in the right direction or the wrong direction with probability of 0.5. For example, if an erroneous bit exists in the ith register, and should be “0” but is wrongly decided as “1”, then a voltage offset $\delta v_n = -2 \cdot DFE_i$ is introduced. DFE therefore rectifies the ISI of the input bit to the pulse “1” direction instead of the pulse “-1”. This degrades the voltage margin of the pulse “-1” while increasing the voltage margin of pulse “1”. Since the current bit has the probability of 0.5 to be “1” or “0”, the raw BER degrades or improves by $\delta_n BER$ or $\delta_p BER$ at the probability of 0.5 respectively.

Enhancing BER with Error Correction

As introduced above, high speed serial links have a mixed error mechanism of random errors and burst errors. There are two broad categories of error correcting codes that are commonly used; block codes and convolution codes. Among the former, cyclic codes have appeared to have great application potential in high speed serial links due to their effectiveness and easy implementation.

According to 802.3ap and CEI2.0 standards, two kinds of cyclic codes are recommended for correcting random and single burst errors, known as Fire codes. From the BCH category, Reed-Solomon (RS) code has been proven to be a particularly powerful technique in dealing with multiple burst errors, and is worth of studying in backplane engineering applications. In the following section, probability calculation is used to estimate the total link BER and to calculate the effect of the 3 kinds of cyclic codes in terms of enhancing link BER performance.

It should be noted that raw voltage bathtub obtained from the serial link simulation is random in nature, and the burst error effect is not accounted for. We assume Err_DFE is the probability vector of a burst length from error propagation, (Err_DFE contains rll probability values, where rll is the maximum propagation length). Err_rand is the random error rate and N is the packet length. The total BER is then calculated by:

$$BER_{total} = \frac{1}{N}\sum_{allP}\sum_{i=0}^{H} P_{pkt}(i+1|i) \qquad (4)$$

Where, $P_{pkt}(i+1|i)$ is a function of the vector Err_DFE, and $P_{pkt}(i+1|i)$ stands for the $(i+1)th$ burst error rate in a packet under the condition that the ith burst error occurs in the same packet. When i equals to 0, P_{pkt} is simply the probability vector of the 1st burst error in the packet. H is a customized number that is determined by Err_DFE. H is a reasonably picked threshold value when $\frac{1}{N}\sum_{allP}\sum_{i=0}^{H} P_{pkt}(i+1|i)$ contributes negligible amount to BER_{total}.

Based on the correction capabilities of different error correction codes, different probability levels can be subtracted from BER_{total} to get the enhanced BER values.

Prediction of BER Improvement Using Error Correction Methods

In this section, an engineering procedure is applied to an experimental backplane system, demonstrating an effective way to identify link margin using AMI simulation and to enhance link margin using FEC calculation.

The proposed flow for the incorporation of error correction into serial link analysis is as follows:

1. For the channels of interest, S-parameter models are acquired from measurement using a high bandwidth Vector Network Analyzer (VNA)
2. To produce raw bathtub curves (without error propagation), serial link signal integrity analysis is performed using commercial EDA (electronic design automation) software and IBIS-AMI models
3. Marginal or failing serial links are identified as candidates for FEC
4. Performing error propagation calculation; during this process the converged DFE tap coefficients from simulation are used to compute the burst error voltage offsets, which in turn serves as the basis for FEC performance analysis, producing the burst error order of the serial link
5. Error correction analysis, to predict the BER performance enhancement

The experimental system consists of a backplane and two daughter cards, which are connected through Airmax and Impact connectors. The link insertion loss at the Nyquist frequency (data rate: 10.3125Gbps) varies from 15dB to 27dB with approximate 1dB increments. The link crosstalk is assumed to be dominated by the connectors because of the optimization of traces and vias.

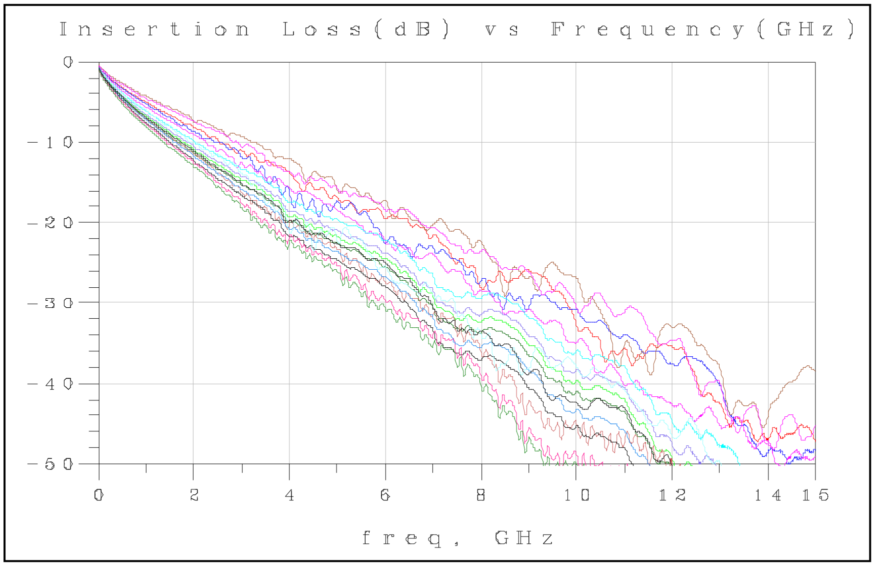

Figure 12a - Insertion loss of the experimental system

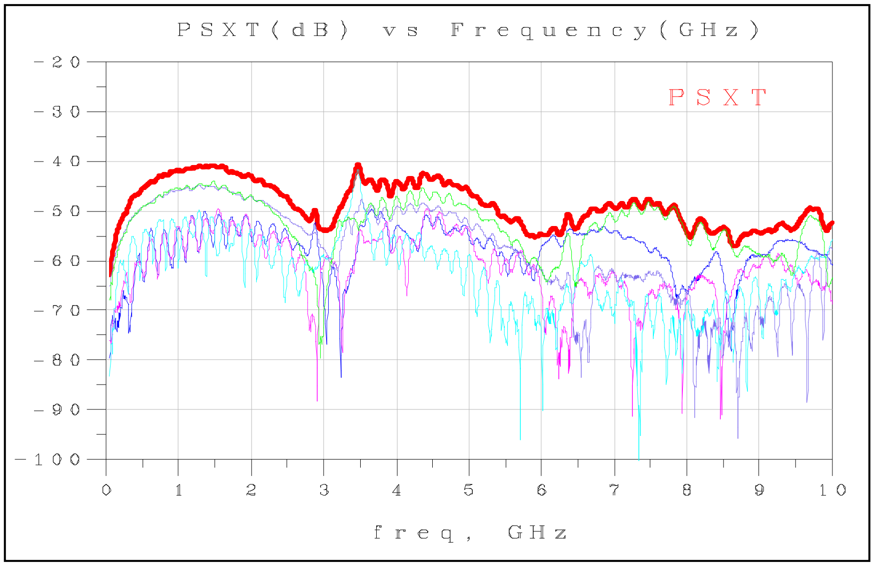

Figure 12b - PSXT (power sum crosstalk) of the experimental system

Serial link margin simulation is performed using HSS12 (High Speed Serial – 12Gbps) AMI models provided by IBM, run at the worst case corner. The targeted BER is 1e-17. The bit stream size is 2000000, using a PRBS23 pattern, with 64B66B coding.

Table 1 - Link margin simulation results

Insertion Loss @ 5.15625GHz	Weighted Eye Height (mVpd)	H_eye(UI) @ 1e-17	V_eye(mVpd) @ 1e-17	Margin Identification
15.288dB	162	0.31	56.4	Good Margin
16.152dB	161	0.31	54.9	Good Margin
17.313dB	155	0.33	55.4	Good Margin
18.208dB	148	0.28	43.4	Good Margin
19.272dB	152	0.28	43.9	Good Margin
20.146dB	147	0.25	36.7	Good Margin
20.627dB	135	0.22	29	Marginal
21.529dB	144	0.22	29.6	Marginal
22.305dB	138	0.27	32.1	Marginal
23.066dB	133	0.21	23.8	Bad margin
23.118dB	135	0.24	26.5	Marginal
24.129dB	119	0.19	18.4	Failed
24.226dB	119	0.18	17.7	Failed
25.293dB	118	0.045	4.62	Failed
26.230dB	111	0.09	8.74	Failed
27.063dB	99.6	0.05	4.45	Failed

As illustrated in Table 1, “Weighted Eye Height” refers to the mean vertical eye opening at the sampling phase inside the eye. Eye width and eye height refer to the post-DFE horizontal and vertical eye openings at the sampling point, respectively.

The ultimate goal for a high speed serial link simulation is to predict whether the link can function properly at the targeted BER. However, there is no specific pass or fail criteria for system simulation, because all impairments of the link can not be accounted for in the early simulation stages. As a result, it is necessary to reserve sufficient margin for unexpected interferences, such as board manufacturing deviations and environmental impacts on the links.

The link margin is dictated by the post-DFE signal quality at the latch. For the IBM HSS12 core, 10~15% UI of horizontal eye opening and 20~30mVpd of vertical eye opening are required. These eye dimensions are required under worst case conditions considering all channel impairments and manufacturing deviations.

According to the criteria above, 10 out of 16 channels are marginal or failing. These 10 channels (labeled in yellow) are selected for performing FEC simulation.

With the computed cumulative BER performances of these channels, including burst errors, FEC algorithms are applied to examine potential improvements in BER. These FEC algorithms include:

- FEC that can deal with multiple random errors (BCH code)
- FEC that can deal with single burst errors (Fire code)
- FEC that can deal with multiple burst errors (RS code)

Each of the FEC codes have different error correcting capabilities, with tradeoffs between capability and complexity. For the FEC codes used in this experiment, after evaluating the implementation complexity, we chose the type of BCH code that has the capability to correct 3 random errors, Fire code that can correct 7 and 11 burst errors respectively, and RS code that has the maximum capability of correcting burst errors for 8 symbols.

For the marginal links, error propagation probabilities are calculated using (3). Figure 13 gives the error propagation probabilities of 4 sample links. Note that the error propagation probability levels are not only related to the DFE coefficients, but also related to the error nature of the links.

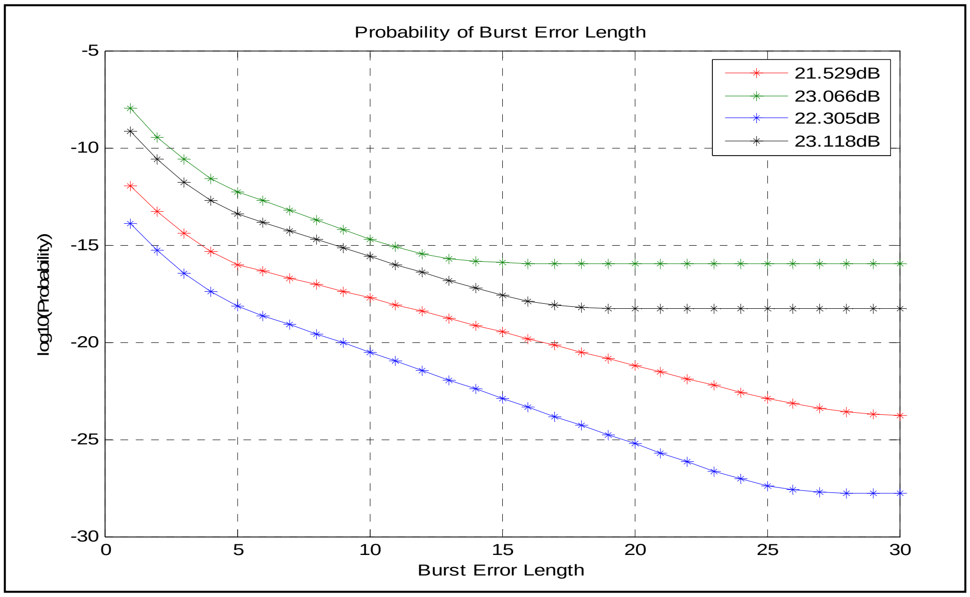

Figure 13 -Error propagation probabilities of the marginal and failing links

With the obtained error propagation probability models, the BER enhancement due to FEC application is calculated.

The FEC simulations were performed with custom-coded MATLAB functions. Because of the statistical nature of the methods used here, each link takes only a couple of minutes for the FEC simulation.

Figure 14 gives the BER levels of the 10 selected links with the 3 kinds of FEC algorithms.

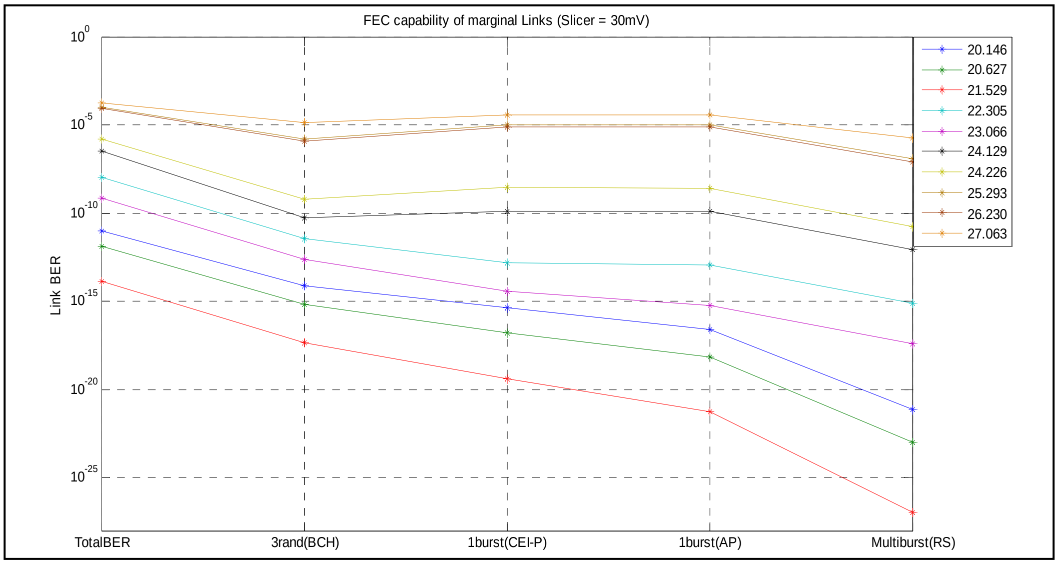

Figure 14 - BER performance enhancement of the FEC codes

As illustrated in the Figure 14, “TotalBER” is the link BER calculated using (4); this value stands for the link BER including both random and burst error impacts. “3rand” is the link BER with the FEC that can deal with 3 random errors. “1burst” stands for the link BER with the FEC that can deal with 1 burst error, where the burst length is 7 (CEI-P) and 11 (802.3AP) respectively. “Multiburst” is the BER for the link using the FEC that can correct 8 erroneous symbols.

Table 2 - FEC simulation results

Total BER	**3rand Corrected**	**1burst corrected (CEI-P)**	**1burst corrected (AP)**	**8symbol corrected**
1.00000e-011	7.73003e-015	4.03629e-016	2.52970e-017	7.00002e-022
1.21153e-012	6.46894e-016	1.62794e-017	6.68200e-019	1.02739e-023
1.25893e-014	4.70710e-018	3.93641e-020	5.11381e-022	1.10441e-027
1.05249e-008	3.33812e-012	1.43592e-013	1.16907e-013	7.75450e-016
7.19685e-010	2.38470e-013	3.49405e-015	6.03374e-016	3.62564e-018
3.35187e-007	5.10521e-011	1.18689e-010	1.18198e-010	7.87583e-013
1.60260e-006	6.28855e-010	2.70849e-009	2.69767e-009	1.81018e-011
1.03594e-004	1.68044e-006	1.00548e-005	1.00096e-005	1.14829e-007
8.80513e-005	1.15447e-006	7.38548e-006	7.35295e-006	7.78913e-008
1.87017e-004	1.40170e-005	3.58155e-005	3.52769e-005	1.85223e-006

From the simulated data shown above, it can be concluded that for the marginal links and the particular FEC codes used here:

- The 3-random correcting BCH can improve the BER approximately by 10^3;
- The 1-burst error correcting Fire codes can improve the BER by at least 10^4;
- The RS code (correcting 8 symbols) can achieve a BER enhancement of 10^8.

However, FEC does not appear to have the ability to transform failing links into those with good margin. Despite the potential improvement with FEC in place, both simulation and experiments to date have shown that certain bad channels will fail their associated BER requirements even with FEC on, such as those in this experiment with insertion losses in the 23dB to 27dB range. These initial trials indicate FEC may not be a panacea, but may be an effective way to improve the BER margin for marginal channels in existing hardware.

Conclusions

This paper has presented a methodology that can be used to quantify BER enhancement in marginal multi-gigabit serial links using error correction codes. This methodology is achieved through the combination of channel simulation, AMI modeling, and statistically-based FEC capability simulation algorithms. The starting point is channel simulation, from which noise bathtub and DFE coefficients are obtained. A “slicer” algorithm is then utilized together with the DFE coefficients to calculate error propagation. The method used here enables the interoperability of noise bathtub and FEC simulation.

Due to the statistical nature of the post processing method used with AMI simulation and the probability calculation method of FEC, an entire end-to-end FEC simulation will take only a couple of minutes. This is proven to be an efficient way to evaluate the potential benefits of applying FEC capability to serial links

In this study, FEC is shown to be very effective in enhancing the BER performance of marginal links. The enhancement can be as much as 8 orders of magnitude. However, FEC does not appear to have the ability to transform failing links into those with good margin.

References

[1] Ransom Stephens, “Jitter analysis: The Dual-Dirac Model, RJ/DJ, and Q-scale, Version 1.0”, Agilent Technologies, 31-December-2004.
[2] Mike Peng Li, Jitter, Noise and Signal Integrity at High-Speed, Prentice Hall 2008.
[3] Cathy Ye Liu and Joe Caroselli, “Modeling and Mitigation of Error Propagation of Decision Feedback Equalization in High Speed Backplane Transceivers.” Proceedings of DesignCon 2006.
[4] Anthony Sanders, “DFE Error Propagation and FEC Comparisons”, OIF2003.245.01, 2003.
[5] Shu Lin and Daniel J. Costello, Error Control Coding: Fundamentals and Applications, Prentice Hall, 2002.
[6] IBM, “HSSCDR User’s Guide”, 2008.

Acknowledgements

Special thanks to Kelly Willis for her assistance in the formatting of this paper.