DESIGNCON® 2025
WHERE THE CHIP MEETS THE BOARD

# Post-FEC BER Analysis of 200 Gb/s Wireline Systems using an FPGA Platform

Richard Barrie, University of Toronto
richard.barrie@isl.utoronto.ca

Ming Yang, University of Toronto/Alphawave Semi
ming.yang@isl.utoronto.ca

Hossein Shakiba, Huawei Technologies Canada
hossein.shakiba@huawei.com

Anthony Chan Carusone, University of Toronto/Alphawave Semi
tony.chan.carusone@isl.utoronto.ca

## Abstract

As wireline communication links transition from 100 Gb/s to 200 Gb/s per lane, new and complex forward error correction (FEC) architectures have been adopted to achieve low bit error ratios (BERs). It is essential to understand how these architectures impact link performance under various channel conditions. However, software-based time-domain simulations are incapable of demonstrating sufficiently low post-FEC BERs in a reasonable amount of time, and existing statistical BER analysis techniques are incapable of accurately modeling the complex FEC architecture proposed for 200 Gb/s wireline systems. This paper presents a flexible platform for field-programmable gate array (FPGA)-accelerated time-domain BER simulations. The platform includes functionality to model wireline systems relevant to the upcoming 200 Gb/s Ethernet standard that include multi-part links, concatenated FEC, soft-decision (SD) decoding, and convolutional interleaving. This FPGA platform can accurately demonstrate post-FEC BERs at the $10^{-12}$ level within a day of simulation time, a speed improvement by a factor of at least 10,000 over software-based simulation platforms.

## Authors Biography

**Richard Barrie** received his M.A.Sc. degree from the University of Toronto in 2024. His research interests are in system modeling and design for high-speed communications.

**Ming Yang** received his Ph.D. from the University of Toronto in 2024. He works as a systems engineer at Alphawave Semi.

**Hossein Shakiba** received his Ph.D. from the Edward S. Rogers Sr. Department of Electrical & Computer Engineering at University of Toronto in 1997. He has over 35 years of teaching, research, design, and management experience in the area of analog circuit and system design for various applications with focus on wireline communication in both industry and academia. He is currently with Huawei Technologies Canada and collaborates with the wireline industry for development of modeling, analysis, and standardization of emerging links. Dr. Shakiba has been the recipient of several awards, including the 1999 IEEE Darlington Award.

**Anthony Chan Carusone** received his Ph.D. from the University of Toronto in 2002 and has since been a professor in the University of Toronto's Department of Electrical and Computer Engineering. He is currently the Chief Technology Officer of Alphawave Semi.
Prof. Chan Carusone co-authored Best Student Papers at the 2007, 2008, 2011, and 2022 Custom Integrated Circuits Conferences, the Best Invited Paper at the 2010 Custom Integrated Circuits Conference, the Best Paper at the 2005 Compound Semiconductor Integrated Circuits Symposium, the Best Young Scientist Paper at the 2014 European Solid-State Circuits Conference, and a Best Paper at DesignCon 2021 and 2023. He co-authored the popular textbooks "Analog Integrated Circuit Design" (along with D. Johns and K. Martin) and "Microelectronic Circuits," 8th edition (along with A. Sedra, K.C. Smith and V. Gaudet). He was Editor-in-Chief of the IEEE Transactions on Circuits and Systems II: Express Briefs in 2009, and an Associate Editor for the IEEE Journal of Solid-State Circuits 2010-2017. He was a Distinguished Lecturer for the IEEE Solid-State Circuits Society 2015-2017. He was the Editor-in-Chief of the IEEE Solid-State Circuits Letters from 2021-2023. Prof. Chan Carusone is an IEEE Fellow.

# 1. Introduction

## 1.1 Motivation

The demand for wireline interconnectivity is rapidly growing, driven by AI applications, video streaming, and cloud services. Forward error correction (FEC) codes are necessary for modern high-speed wireline links to meet stringent bit error ratio (BER) requirements. For wireline links that operate up to 100 Gb/s per lane using four-level pulse amplitude modulation (PAM-4), a single end-to-end FEC code, as shown in Figure 1, can achieve a sufficient post-FEC BER with acceptable latency and power. The next generation of PAM-4 wireline links will operate at 200 Gb/s per lane. Doubling the data rate degrades signal integrity, necessitating a stronger FEC architecture. To address this, the IEEE 802.3dj task force has adopted a new concatenated FEC architecture for the 200 Gb/s Ethernet standard. A concatenated FEC has an inner code and an outer code that provide two layers of error correction, as shown in Figure 2. A soft-decision decoder may be used to boost the performance of the inner code, while a hard-decision decoder is used for the outer code.

Many factors impact the post-FEC BER of a wireline system, including the choice of FEC codes and decoding algorithms, signal-processing algorithms such as precoding and interleaving, analog and digital equalization, noise [1], and other channel impairments. Tools that help us understand precisely how these factors affect a system's performance are useful for two purposes: firstly, they are applicable in developing communication protocol standards that specify the FEC architecture and performance requirements; secondly, they are helpful for system architects to know how their design decisions affect the overall performance of a wireline communication system.

Standardized communication protocols such as Ethernet, PCIe, and USB are crucial to ensure the interoperability of products between different vendors. Currently, the 200 Gb/s Ethernet standard is still a work in progress. Industry-leading companies are heavily investing in the development of 200 Gb/s SerDes products to meet the global demand for wireline connectivity. These companies come together regularly during IEEE 802.3dj task force meetings to define the standard with which their products must comply. Many details of the system-level architecture and FEC performance are still up for debate, creating a need for tools that can analyze the post-FEC BER of 200 Gb/s systems [2].

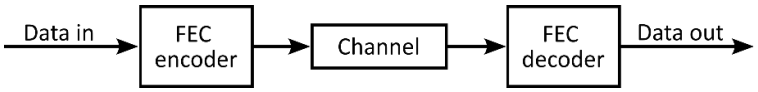

Figure 1. End-to-end FEC

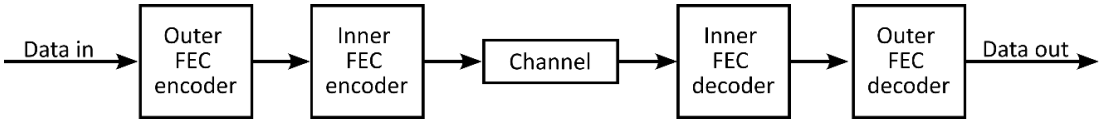

Figure 2. Concatenated FEC

One option for analyzing a 200 Gb/s wireline system is to perform a time-domain software simulation by sending data through a model of the wireline system and counting the number of post-FEC bit errors. However, time-domain software simulations demonstrating the extremely low ($< 10^{-13}$) post-FEC BERs targeted by IEEE standards are prohibitively time-consuming, especially for exploring design alternatives. Another possibility is to use a statistical analysis tool to predict a system's post-FEC BER without having to run a long time-domain simulation. However, there are currently no statistical methods available that are capable of realistically modelling the complex architectures considered for the 200 Gb/s Ethernet standard.

This paper presents an FPGA platform as a tool for analyzing the post-FEC BER of wireline systems down to very low levels. It specifically targets the FEC architectures considered for the upcoming 200 Gb/s Ethernet standard. This paper aims to give the reader an understanding of the various wireline systems that can be modelled on the FPGA platform. It also presents simulation results and analysis. For readers who wish to replicate or build on the platform presented in this document, Verilog code for the FPGA platform is included in the online repository at https://github.com/richard259/FPGA-FEC.

### 1.2 Outline

This paper is organized into eight sections. Section 2 presents background information on the wireline system architecture considered for the upcoming 200 Gb/s Ethernet standard. Section 3 presents the high-level architecture of the proposed FPGA platform. Sections 4 to 6 present how key building blocks in a wireline system are modelled on the FPGA platform. Specifically, Section 4 presents wireline systems based on channel models that artificially inject errors into a stream of PAM-4 symbols. These are used primarily for hard-decision electrical links. Section 5 presents a more complex model for the channel and receiver that includes a discretized approximation of the analog signal, allowing for the modelling of maximum-likelihood sequence detection (MLSD) and soft-output optical receivers. In Section 6, the inner-FEC code is introduced, as well as the Chase algorithm for soft-decision inner-FEC decoding and the generation of soft-decision channel outputs. Simulation results are presented in all these sections, with a comparison to statistical or software models to verify the accuracy when possible. Section 7 presents a case study where a full multi-part 200 Gb/s wireline system is modeled. The FPGA platform is used to analyze the system's performance under various channel conditions and architecture choices, demonstrating the platform's utility in system-level transceiver design and standard development. Finally, Section 8 concludes this paper.

## 2. Background

### 2.1 FEC Architecture for 200 Gb/s Multi-Part Links

Multi-part links are typically employed for communication between data centers. First, a short electrical link transmits data from a host chip on a server to an optical module. The optical module converts this electrical signal to an optical signal and sends it along a long fiber-optic cable to a second optical module at a receiving data center. This optical module converts the signal back to an electrical signal and sends it to a receiving host chip over another electrical link. Both the electrical and optical links use PAM-4 signaling. All three links along the data path can introduce errors in the transmitted data.

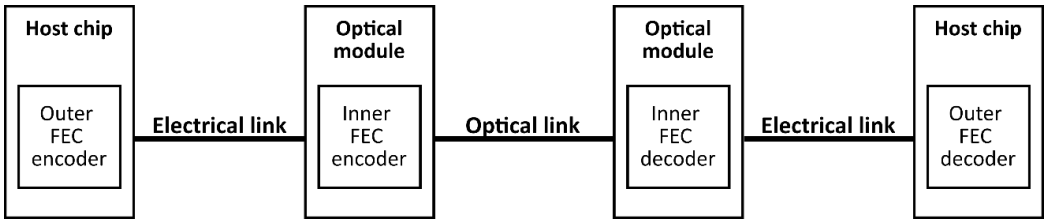

Figure 3. 200 Gb/s multi-part link with concatenated FEC [3].

The electrical links can have large amounts of inter-symbol interference (ISI), so equalization techniques such as decision feedback equalization (DFE) or MLSD are used to recover the signal. Both DFE and MLSD suffer from error propagation, which causes the errors from electrical links to be predominantly bursts of correlated errors [4-5]. In contrast, optical channels are theoretically ISI-free, and are often modelled with an additive-Gaussian-white-noise (AWGN) channel, which introduces uncorrelated random errors. However, in reality, PAM-4 optical receivers typically include bandwidth-limited components such as transimpedance amplifiers that introduce ISI, requiring equalization in optical receivers. Although error propagation may also introduce correlated errors in an optical link, they will typically be less strongly correlated compared to the burst errors in electrical links [6].

For 200 Gb/s multi-part links, the concatenated FEC architecture proposed in the Ethernet standard [7] is shown in Figure 3. The outer code corrects errors in all three links, while the inner code only protects the optical link. The outer code is a Reed-Solomon code with block length $n = 544$ and dimension $k = 514$ in the Galois Field GF($2^{10}$) known as the "KP4" code. The KP4 code is a strong, non-binary linear block code that operates on 10-bit-long FEC symbols. It can correct up to $t = 15$ FEC symbols per 544-FEC-symbol-long codeword, making it well-suited to correct the long bursts of errors that arise in electrical links due to DFE or MLSD error propagation. The inner code is a binary extended Hamming code that can correct $t = 1$ bit error per 128-bit codeword. Although this code can only correct one bit error per codeword, it is effective against uncorrelated errors that arise in optical links and has low decoding complexity compared to a stronger code. Interleaving and soft-decision decoding can protect the inner-FEC decoder from correlated errors in the optical link, thus improving the inner FEC's coding gain. The inner code primarily focuses on correcting random errors in the optical link, so that the outer code can work more effectively on correcting the correlated burst errors.

## 2.2 System Performance Metrics

The post-FEC BER, defined as the ratio of bit errors after FEC decoding to total transmitted bits, is a commonly used measure of communication reliability. However, in the context of the Ethernet protocol, the codeword error ratio (CER), defined as the ratio of uncorrectable KP4 codewords to the total transmitted KP4 codewords, is a more relevant metric and used in this paper. The current draft of the 200 Gb/s Ethernet standard requires a CER of less than $1.45 \times 10^{-11}$.

## 2.3 Related Works

There are several approaches for estimating the post-FEC BER of a wireline system. One approach is using a software-based time-domain simulation. Unfortunately, software simulations demonstrating BERs below $10^{-12}$ are prohibitively time-consuming [8]. Various extrapolation methods are also used to estimate the post-FEC BER from a short time-domain simulation. In many cases this approach is inaccurate because the correlation between signal-to-noise ratio (SNR) and post-FEC BER may change at lower BERs [8]. A second approach is to use a statistical model to determine post-FEC BERs analytically [8-9]. While statistical models can quickly determine arbitrarily low BERs, they are limited by the complexity of details in the architecture that can be modeled. For example, a statistical model for concatenated FEC codes is proposed in [10], but it cannot capture the complex interleaving and symbol multiplexing schemes in 200 Gb/s systems.

Using an FPGA platform for FEC analysis is not a new idea. It was first proposed in 2002 [11]. Since then, FPGAs have often been used to prototype FEC architectures and run BER analysis simulations [12-13], however there are no existing platforms that are capable of realistically modelling multi-part 200 Gb/s links. Another feature that sets this work apart is that other FPGA platforms typically assume an AWGN channel. In contrast, this work can accommodate a more complex system including the modelling of multi-part links, channels with ISI, interleaving, precoding, DFE and MLSD-based receivers, and colored noise.

# 3. FPGA Platform Overview

Simulation results presented in this paper were run on a single Xilinx Virtex UltraScale FPGA VCU108 evaluation kit, however the FPGA platform can be adapted for use on other hardware. Figure 4 shows a system-level diagram of the FPGA simulation platform. The first step in running a simulation is to model the system of interest by defining the "BER simulation IP" block. This block is defined using a library of custom Verilog modules. This library is implemented in a flexible way so that one can mix and match modules to model a wide variety of wireline systems. Many different channel types, FEC architectures, and interleaving schemes can be included as part of the transceiver model.

Once the BER simulation IP block is defined, the FPGA is programmed with the system shown in Figure 4. The user configures simulation settings with an interface on a PC. Software running on the on-chip processor configures the BER simulation IP with these settings and starts the simulation. The BER simulation IP generates binary data and sends

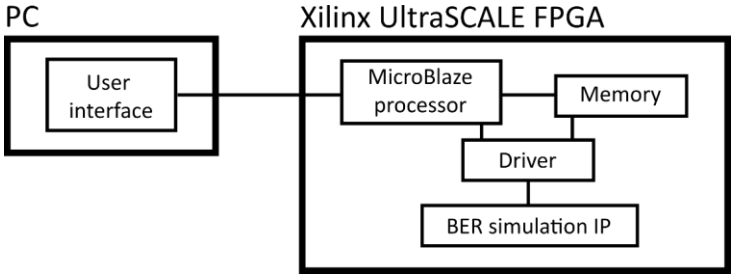


Figure. 4 FPGA-accelerated simulation platform [3].

it through the wireline system model. As data is sent through the system model, the following statistics are counted:

- Number of transmitted bits
- Number of bit errors before KP4 decoding (pre-FEC bit errors)
- Number of bit errors after KP4 decoding (post-FEC bit errors)
- Number of transmitted KP4 codewords
- Number of KP4 codeword errors

Once a user-specified number of KP4 codeword errors is reached, the results are automatically written to the memory and can be viewed on the user interface and saved to a .csv file. The software running on the processor allows the user to instantiate and run multiple consecutive simulations automatically. This way, the platform can efficiently sweep link SNR over a range of values to view its relationship with CER. Configurability is one of the platform's main assets, allowing it to explore different FEC architectures and DSP algorithms under various channel conditions. There are three levels at which a user can customize the BER simulation IP:

1. The **system architecture** is defined in a top-level Verilog file. This file specifies all the Verilog modules that make up the wireline system model. For example, this file determines if the system is a single-link or multi-part link, the type of channel models used, if an inner FEC code is used, etc... The system architecture is the least flexible level of customization. To change the system architecture, a user must edit the top-level Verilog file, re-synthesize the hardware and re-program the FPGA.
2. The Verilog **module parameters** are used to customize Verilog modules. These parameters include the number of bits used to represent an analog or digital signal, the value of $\alpha$ in a $1+\alpha D$ channel pulse response, and the number of parallel cores used to speed up the simulation. Changing these parameters is more accessible than changing the system architecture because they can be set in the block design GUI without editing a Verilog file. However, changing these parameters requires re-synthesizing the design and re-programming the FPGA.
3. The **runtime-programmable settings** are the most flexible settings. They can be changed with the user interface after the FPGA is programmed. These settings include the SNR of each link, inner FEC on/off, precoding on/off, and the number of codeword errors to observe before terminating the simulation.

## 3.1 Simulation Speed

With the VCU108 evaluation kit used in this paper, The platform runs at 150 MHz, with one bit per clock cycle simulated, resulting in a simulation speed of 150 million bits per second. The platform also supports parallel processing, where the BER simulation IP contains multiple parallel cores to increase simulation speed. The number of parallel cores that can be used depends on the hardware complexity of the system of interest. To achieve maximum speed, the number of cores should be set to the highest amount possible subject to the available hardware resources on the FPGA and additional timing constraints. For simulations of the simplest channel model including a single-part link with AWGN, the hardware used in this paper can support 200 parallel cores for a

simulation speed of 30 Gb/s. For a more complex system including a multi-part link with soft-decision inner FEC decoding, the hardware complexity of the BER simulation IP is significantly higher. Accordingly, only 8 parallel cores can be used, resulting in a slower simulation speed of 1.2 Gb/s.

### 3.2 Confidence Intervals on CER Estimates

When performing a time-domain simulation, an estimate of the CER, $CER_{est}$, is calculated as the ratio of uncorrectable codewords to total transmitted codewords. The event that each transmitted KP4 codeword is either correctable or uncorrectable can be viewed as a binary random variable. Codewords with 15 or fewer FEC-symbol errors are correctable, and those with greater than 15 are not. As the number of transmitted codewords approaches infinity, the ratio of codeword errors to total codewords will approach the true CER of the system, $CER_{true}$. However, the simulation must be terminated after some finite number of codeword errors. Terminating the simulation after a smaller number of transmitted codewords results in a shorter simulation with lower accuracy, whereas terminating after a larger number of transmitted codewords results in better accuracy at the cost of longer simulation time. It is helpful to calculate the confidence intervals on CER estimates and choose a simulation termination condition with acceptable accuracy in the shortest time. One can approximate the event of correctable or uncorrectable codewords as an independent and identically distributed (i.i.d) Bernoulli process so that the Clopper-Pearson method [14] can be used to calculate the confidence intervals of a CER estimation. For the simulations of low ($< 10^{-8}$) CERs presented in this paper, the simulation termination condition is chosen to be 20 observed codeword errors, giving a 90% confidence that $CER_{est}$ is within -33% to +47% of $CER_{true}$. For example, if $CER_{est} = 1.45\times10^{-11}$ with 20 observed codeword errors, this results in a 90% confidence interval that $CER_{true}$ is from $9.6\times10^{-12}$ to $2.1\times10^{-11}$.

## 4. Wireline Systems with Error Injection Channel Models

This section describes the FPGA platform's implementation of data generation, PAM-4 modulation, channel models, and the KP4 FEC. This chapter focuses on simple channel models that artificially inject errors into a stream of PAM-4 symbols. These "error injection" channel models have the advantage of a low hardware complexity and can accurately represent the errors introduced by an AWGN channel and channels with burst errors.

### 4.1 Input PRBS Data, PAM-4 Modulation, and Precoding

Pseudo-random binary sequences are commonly used as input data for BER testing because they exhibit statistically random behavior and are easily generated with linear-feedback shift register circuits. For all simulation results shown in this paper, we use a PRBS-63 sequence. The FPGA platform also supports the industry standard PRBS-31 pattern. Grey-coded PAM-4 modulation is applied to the input binary sequence before transmission through electrical and optical channels. Our platform also has modules implementing 1/(1+D) precoding, which is a DSP algorithm used to reduce the impact of bursts of consecutive PAM-4-symbol errors that arise in electrical links due to DFE or MLSD error propagation [15]. 1/(1+D) precoding eliminates most errors from a burst, leaving only the first and last symbol errors.

## 4.2 Outer-FEC Code Model

To avoid a complex and resource-intensive RS KP4 encoder and decoder implementation, the encoder and decoders are not included in the FPGA model. Instead, the outer RS FEC code implementation is simplified by counting the number of FEC-symbol errors using a PRBS checker. The same PRBS pattern can be generated and then compared with the data at the system's output. Pre-FEC bit errors are identified if the output data does not match the PRBS. This concept can be extended to check the post-FEC BER/CER as well. To do this, a "FEC checker" module checks each group of 10 consecutive bits at the system's output against the reference PRBS. A FEC-symbol error is identified if there are one or more bit errors in a 10-bit-long sequence. The FEC checker tracks the number of bit errors and FEC-symbol errors for each KP4 codeword. If there are more than 15 FEC-symbol errors at the end of a 544 FEC-symbol-long KP4 codeword, the codeword is not correctable, and a codeword error count increases by one. All the bit errors in the uncorrectable codeword are added to the count of post-FEC bit errors. Otherwise the codeword is correctable, and no codeword error or post-FEC bit errors are counted. The hardware implementation of this FEC checker module has Verilog parameters to set the FEC code parameters $n$, $k$ and $t$, as well as the number of bits per FEC symbol so that it can model any linear block code. 64-bit unsigned values are used for all counters so they will not overflow during long simulations.

The FEC checker can optionally output and save the number of codewords having all numbers of FEC symbol errors: (0,1,2,…,15,16+) with some increased routing complexity for the additional 64-bit counters. This can be used to construct block error histograms as described in Annex 174A of the IEEE 802.3dj standard.

## 4.3 Block Interleaving

Block interleaving mitigates the impact of burst errors on FEC performance [16]. With $N$-way block interleaving, $N$ different codewords are transmitted through the channel simultaneously, with their FEC symbols interleaved in a round-robin fashion. An example of 2-way block interleaving is shown in Figure 5.

This architecture distributes consecutive FEC-symbol errors over multiple codewords, giving them a greater chance of being corrected by the KP4 decoder. Both 2-way and 4-way block interleaving schemes are considered for 200 Gb/s applications [17]. Our FPGA platform supports $N$-way interleaving schemes by making a simple modification to the KP4 checker module: the module keeps track of which of the $N$ codewords the current FEC symbol belongs to and maintains bit errors and FEC-symbol error counts for each of the $N$ codewords individually. The value of $N$ for the KP4 FEC checker module is implemented as a runtime-programmable setting.

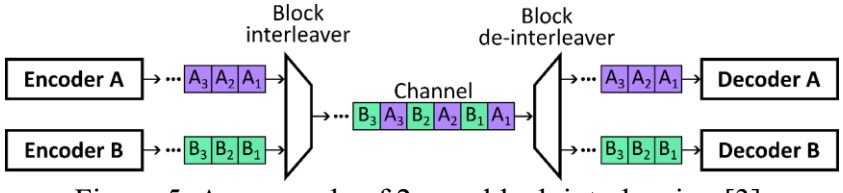


Figure 5. An example of 2-way block interleaving [3].

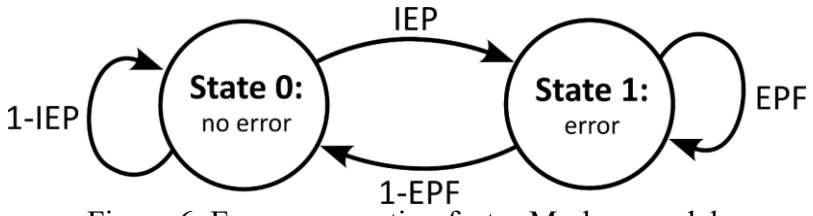

Figure 6. Error propagation factor Markov model.

## 4.4 Error Propagation Factor (EPF) Channel

The error propagation factor (EPF) channel model is a two-state Markov model for modelling DFE and MLSD error propagation proposed in an Ethernet standard contribution [5]. Figure 6 shows the state diagram of the Markov model modelling error propagation.

Visiting the no-error state '0' at time *k* signifies that there is no PAM-4 symbol error occurred and visiting the error state '1' signifies that there is an error. Given that the current state is the no-error state '0', the probability of transitioning to the error state '1' is the initial error probability (IEP). Given that the current state is the error state, the probability of having a subsequent PAM-4 symbol error is defined as the error propagation factor (EPF). For consecutive PAM-4 symbol errors, the error sequence alternates between +1 and -1 to mimic the behavior of DFE and MLSD error propagation, so precoding is effective against the artificial errors injected by the EPF channel model [8]. These two parameters, IEP and EPF, entirely define the behavior of the EPF channel model.

### 4.4.1 FPGA Implementation

The IEP and EPF parameters are converted to a 64-bit representation where 0 represents zero probability, and $2^{64}$-1 represents a probability of 1. These parameters are runtime-programmable settings that can be loaded into the EPF channel module. The module also contains a register to track the current Markov state. A 64-bit uniform random number generator (URNG) realizes the random transitions between the error and no-error states based on the EPF and IEP parameters. Errors are injected according to the current Markov state by perturbing the transmitted symbols to one of the nearest symbol levels.

### 4.4.2 Simulation Results

Figure 7 shows pre-FEC BER vs. IEP for an EPF channel. Two types of EPF are considered, a random-error channel with EPF = 0 and a burst-error channel with EPF = 0.75. The accuracy of the FPGA implementation is verified by the analytically determined BER from the statistical model reported in [8], shown with the solid lines. The system-level diagram of the BER Simulation IP programmed on the FPGA is depicted at the top of the figure, and a similar system-level diagram is included for all figures that present FPGA-simulation results in the rest part of this paper.

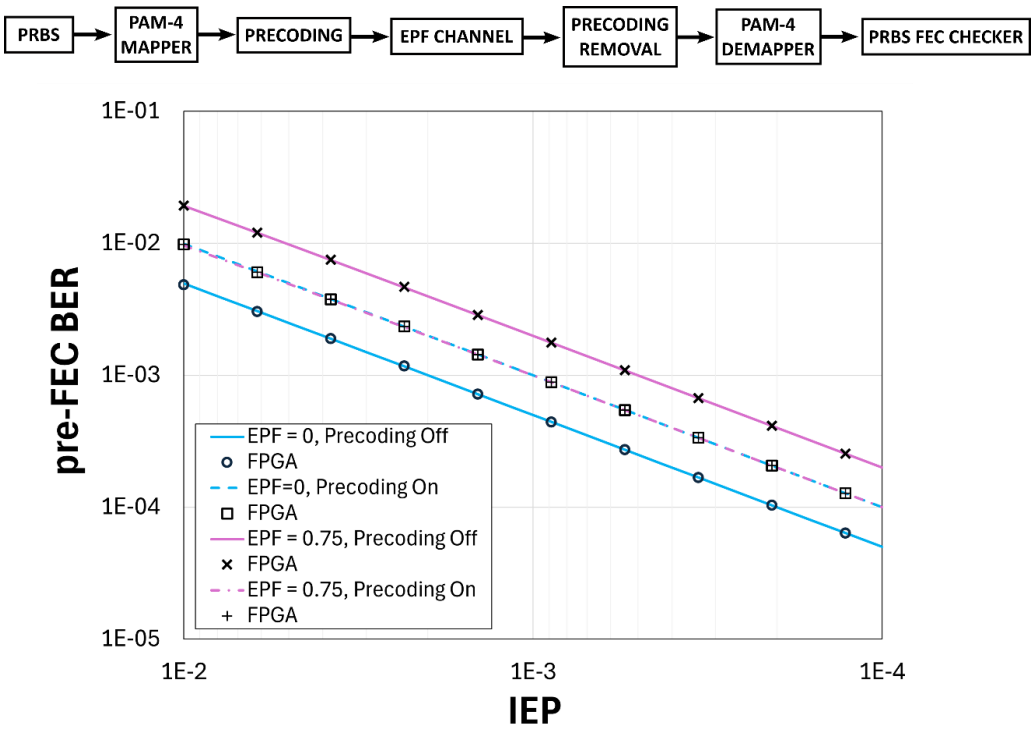


Figure 7. Pre-FEC BER vs. IEP for EPF channels with and without precoding.

For the channel with EPF = 0 and precoding turned off, the pre-FEC BER is about half of the IEP because each PAM-4 symbol error only produces one-bit error. In contrast, with precoding, the pre-FEC BER is the same as the IEP because the precoding turns a random PAM-symbol error into two PAM-symbol errors, each containing one bit error. The pre-FEC BER is much higher for the link with EPF = 0.75 and precoding off because each initial error may result in a long burst. With precoding turned on, most errors in a burst are eliminated, resulting in much lower pre-FEC BERs. As the precoding always turns any random error or a DFE/MLSD burst error into two errors, with precoding turned on the same IEP correspond to the same pre-FEC BER level, regardless of the EPF value.

Figure 8 shows the CER vs. IEP with the same channel conditions as in Figure 7. The link with EPF = 0 and precoding off performs the best because this combination results in the lowest possible pre-FEC BER. The link with EPF = 0.75 and precoding off has the highest CER due to uncorrectable burst errors. The burst errors degrade FEC decoding performance, which can be seen in the slope of the CER vs. IEP plot, flattening at low IEP, an effect known as the “error floor”. Despite the two EPF levels having the same pre-FEC BER with precoding on, the link with EPF = 0 has a lower CER due to the large spacing between random PAM-symbol errors. Given a single PAM-4 symbol error, the precoding will produce an additional symbol error right next to the existing error where the two PAM-4 symbol errors are likely contained in one 10-bit-long FEC symbol. In contrast, the precoding removes most errors in a burst, leaving two errors that may be further spaced apart, making it more likely corrupting multiple FEC symbols.

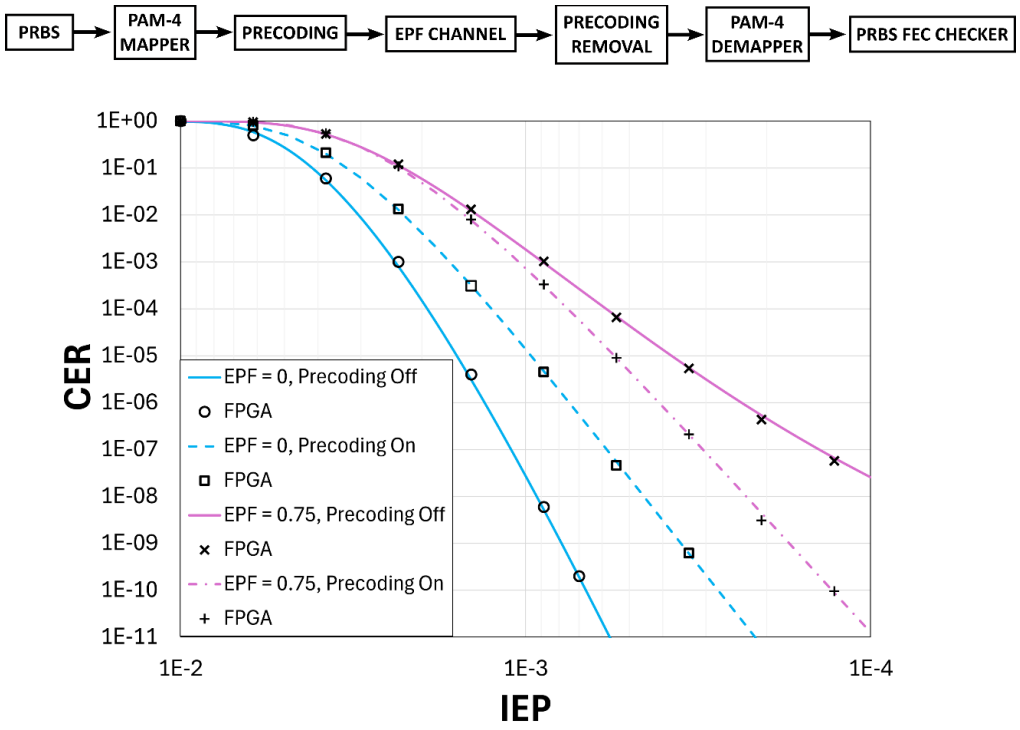


Figure 8. CER vs. IEP for EPF channels with and without precoding.

### 4.5 Other Channels

The FPGA platform also includes error injection channel models that are capable of accurately modeling AWGN channels and $1+\alpha D$ partial-response channels in the presence of AWGN with a zero-forcing DFE.

## 5. Analog Channel and Receiver Models

Section 4 presented channel models that artificially inject errors into the stream of transmitted PAM-4 symbols. Although these models have a relatively low hardware complexity, they have some shortcomings. Firstly, they cannot accurately model a MLSD-based receiver, which is a key equalizer block for 200 Gb/s applications. Secondly, they are not well-suited to produce the soft reliability information required for soft-decision inner-FEC decoding because they do not include any representation of a continuous analog signal or time-domain noise signal. This chapter presents a more complex channel model to address these shortcomings. instead of simply injecting artificial errors into a stream of PAM-4 symbols, each PAM-4 symbol is now represented as a discretized analog level using a fixed-point signed decimal representation. This baud-rate signal may include a $1+\alpha D$ response that adds ISI. A time-domain noise signal in fixed-point decimal representation is generated on the FPGA and added to the signal. The output symbols may be recovered by a simple ideal slicer or an MLSD or DFE-based equalizer (if ISIs are present in the channel). Next, Section 6 builds on this analog channel model by introducing models of soft-output receivers that generate the reliability information required for soft-decision decoding.

## 5.1 Analog AWGN Channel

Figure 9 (a) shows a block diagram of an ideal PAM-4 AWGN channel, and Figure 9 (b) shows its hardware implementation on the FPGA. The PAM-4 symbols $b_k$ are mapped to the transmitted signal levels $t_k^{bin}$ in an $m$-bit signed decimal representation. The superscript "*bin*" denotes that $t_k^{bin}$ is the FPGA platform's binary representation of the analog signal level $t_k$. The Gaussian noise samples $n_k^{bin}$ are then added to $t_k^{bin}$. The signal is clipped to maintain the m-bit resolution for the received noisy samples $r_k^{bin}$. With an appropriate choice of signal resolution, this signal clipping does not affect the PAM-4 symbol error rate or the post-FEC BER.

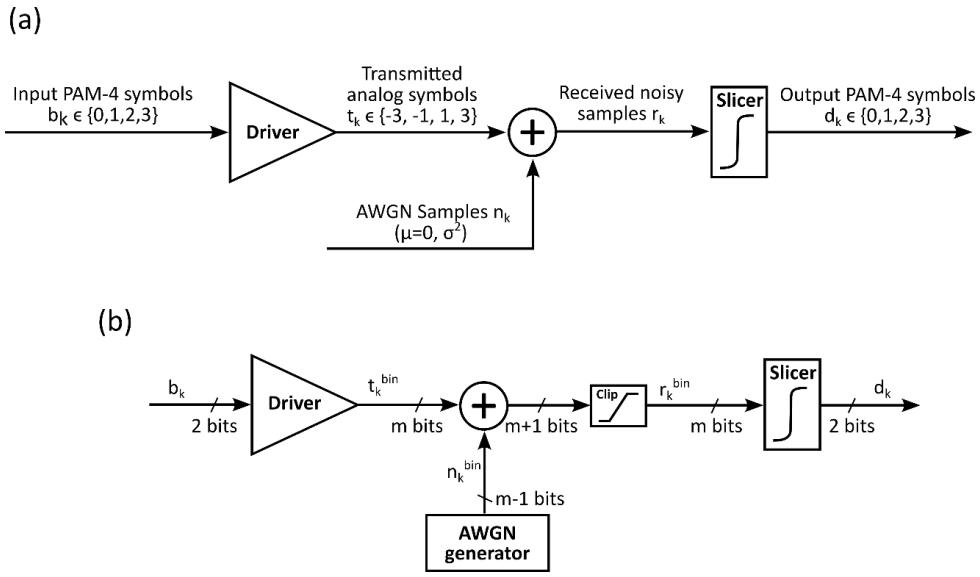


Figure 9. (a) Analog PAM-4 channel with AWGN. (b) FPGA implementation [3].

An ideal slicer outputs the most likely output signal $d_k$. A Verilog parameter $m$ controls the number of bits representing the analog signal. Increasing $m$ produces a finer resolution that may match better to a statistical or software model that assumes a floating-point representation of analog signals. However, increasing $m$ comes at the cost of additional hardware complexity. A difference between this model and a real 200 Gb/s receiver is that the FPGA platform has no analog-to-digital converter (ADC). Instead, the resolution $m$ is assumed to be the same as the ADC resolution of interest. This way, the same signal resolution can be used throughout the model, reducing hardware complexity. An eight-bit signal resolution is chosen for a realistic ADC in a 200 Gb/s receiver, and so $m$=8 is used for all simulation results shown in this paper.

## 5.2 AWGN Generator

There are multiple approaches for generating AWGN samples on an FPGA; for example, the Box-Muller method is commonly used [18]. However, this method is not well-suited for in-runtime programming of the noise variance, which is essential for the platform to efficiently sweep over a range of SNR levels. Instead, the AWGN generator on this FPGA platform uses a pre-computed look-up table that contains the probability for each noise sample, and a URNG to randomly choose noise samples with the correct frequency. In this work, the Tausworthe algorithm was chosen to implement the URNG for its

hardware simplicity and suitable pseudo-random properties. A 64-bit Tausworthe URNG is used to pick the magnitude of each noise sample, and an uncorrelated binary random number generator determines the sign bit.

## 5.3 Slicer

A hard-decision PAM-4 slicer recovers the PAM-4 symbols from the received noisy samples. It calculates the absolute value of the distance of each sample to each of the transmitted symbol levels and outputs the symbol with the lowest distance. A channel output may be equidistant from two of the transmitted symbol levels. In this case, the slicer's tie-breaking rule is always to output the smaller PAM-4 symbol (e.g. if the channel output is equidistant from $b_k = 0$ and $b_k = 1$, the slicer outputs $d_k = 0$). Because all four PAM-4 symbols are equally likely to be transmitted, this results in an error half the time when the channel output is equidistant from two PAM-4 symbols.

## 5.4 Colored Noise

Although a white noise spectrum is commonly used for FEC performance evaluation, real 200 Gb/s links typically experience colored noise due to the frequency-dependent noise amplification of analog equalizers in the system, and bandwidth limitations of the electronic components. The impact of correlated noise samples on FEC performance is not commonly studied, and is not easy to analyze with statistical models. The method for generating colored noise samples on the FPGA platform is to pass the AWGN through an FIR filter. The number of filter coefficients and their values are implemented as customizable Verilog parameters.

## 5.5 1+αD Response and MLSD

The FPGA platform also includes a module to implement an 'analog' 1+$\alpha$D channel that adds ISI, using the signed decimal signal representation introduced earlier in this chapter, shown in Figure 10.

The Viterbi algorithm [18] finds the maximum-likelihood state sequence of a Markov process observed in discrete memoryless noise. This algorithm can be applied to an ISI channel in the presence of AWGN to find the maximum-likelihood sequence of transmitted PAM-4 symbols $b_k$, given the received noisy samples $r_k$. This application is known as maximum-likelihood sequence detection (MLSD), and it offers an inherent advantage over a DFE: while the DFE seeks to do symbol-by symbol detection by subtracting ISI out of each sample $r_k$, MLSD uses the ISI as a useful piece of information to inform its symbol detection over a sequence of symbols.

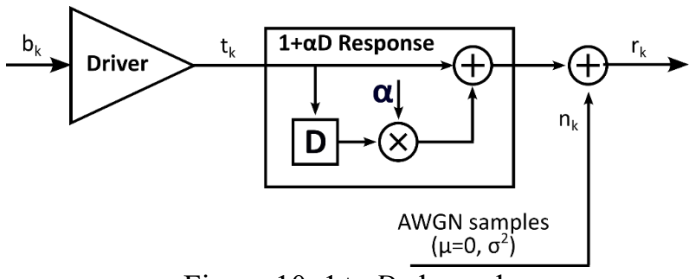


Figure 10. 1+$\alpha D$ channel.

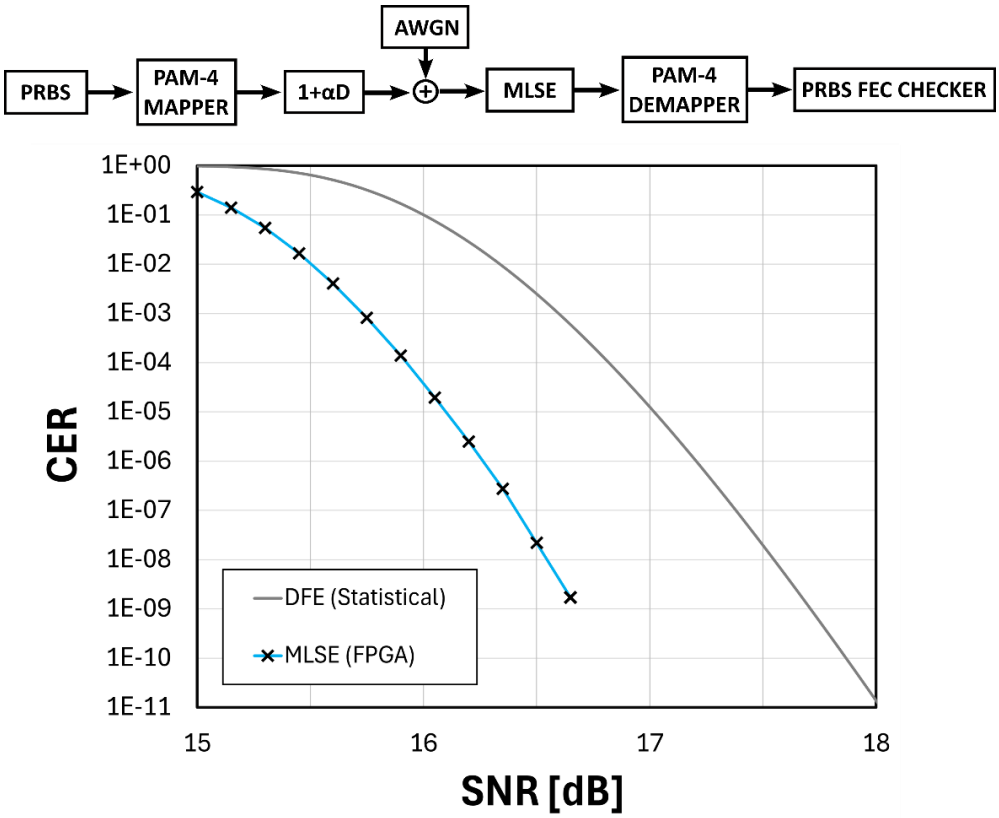


Figure 11. CER vs. SNR 1+0.5*D* channel with AWGN, equalized with MLSD (and DFE for comparison).

Figure 11 shows CER vs. SNR for a 1+0.5*D* channel with AWGN, equalized with MLSD. The same channel equalized with a DFE is included for comparison. The statistical model presented in [8] is used to generate the results for the DFE channel, and the MLSD simulation results are generated on the FPGA platform. MLSD outperforms the DFE across all SNR levels as the large 0.5*D* ISI cursor is part of the signal processed by the MLSD, while the DFE seeks to cancel out the ISI.

# 6. Soft-Decision Inner FEC Code

## 6.1 Hamming (68,60) Code

Since each Gray-coded PAM-4 symbol contains only up to one erred bit [8], the Hamming decoder can correct errors based on PAM symbols instead of bits. A new shortened (68,60) Hamming code has been adopted by the Ethernet standard for use with a soft-decision decoder that protects PAM-4 symbols rather than individual bits [17]. The (68,60) code is achieved by puncturing the extended Hamming (128,120) code in 60 places.

Figure 12 shows the encoding of a shortened Hamming (68,60) codeword. The encoder takes sequences of 120 information bits, denoted $u \in \mathbb{F}_2^{1\times 120}$, and computes codewords $c \in \mathbb{F}_2^{1\times 128}$. The notation $\mathbb{F}_2^{i\times j}$ refers to an $i$ by $j$ vector with entries from $\mathbb{F}_2$ (binary entries). To encode a message, first two bits of each pair that form a PAM-4 symbol in the

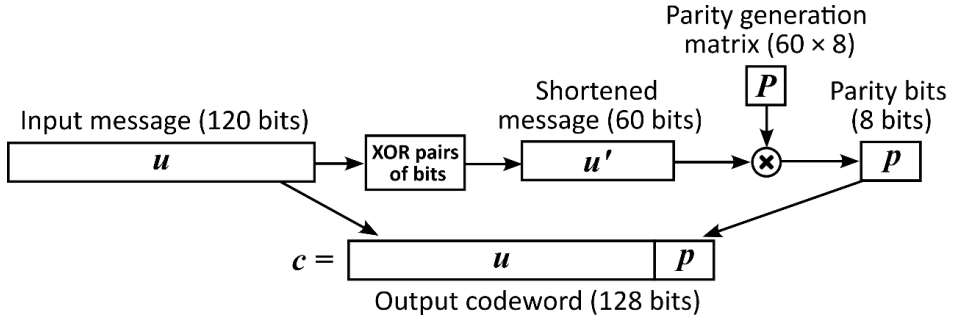

Figure 12. Encoding 120 information bits using the shortened Hamming (68,60) code [3].

message $u$ are XORed together to create a new 60-bit long message $u'$. A sequence of eight parity bits that protects $u'$ is generated with

$$p = u'P \tag{1}$$

where $P \in \mathbb{F}_2^{60\times 8}$ is a new punctured parity generation matrix taken from the draft Ethernet standard [7]. This binary matrix multiplication is a Verilog module that uses XOR gates to perform the binary multiply-accumulate operations between $u$ and the columns of $P$.

Although this (68,60) code is typically decoded with the Chase soft-decision decoding algorithm, we first describe how hard-decision decoding would be done, which is a crucial building block for the Chase algorithm. To begin with, the decoder forms a 68-bit-long received message $y$. The first 60 bits of $y$ are produced by XORing pairs of bits in the first 60 PAM symbols together, and the last eight are produced using both bits from the 4 last PAM-4 symbols.

To decode input messages $y$, first the syndrome $s \in \mathrm{F}_2^{1\times 8}$ is calculated using the parity-check matrix $H \in \mathrm{F}_2^{8\times 68}$

$$s = yH^T \tag{2}$$

Because the shortened Hamming code is systematic, the parity-check matrix $H$ is of the form

$$H = [P^T\ I_8] \tag{3}$$

where $I_8$ is the 8×8 identity matrix. The eight-bit syndrome can now be uniquely mapped to the location of either a PAM-4 symbol error in the 60 PAM-4 symbols or the location of a bit error in the eight parity bits. The syndrome can take $2^8 = 256$ possible values. With the shortened code, only 68 of the 256 possible syndromes point to a valid error location, with another valid syndrome $s = 0$ indicating no error, and the remaining 187 non-zero syndrome values indicate decoding failure.

## 6.2 Convolutional Interleaver

Although the errors introduced in optical links may be predominantly random, the inner-FEC decoder changes the error distribution by correcting the inner-FEC codewords with only one error and not correcting those with multiple errors. As a result, the majority of the 120-bit long inner FEC messages will be error-free. However, occasionally, a codeword will have two or more errors, effectively introducing correlated errors after inner FEC decoding. To give the outer decoder the best chance of correcting errors from an erroneous 120-bit-long inner-FEC codeword, each of its twelve 10-bit-long outer-FEC

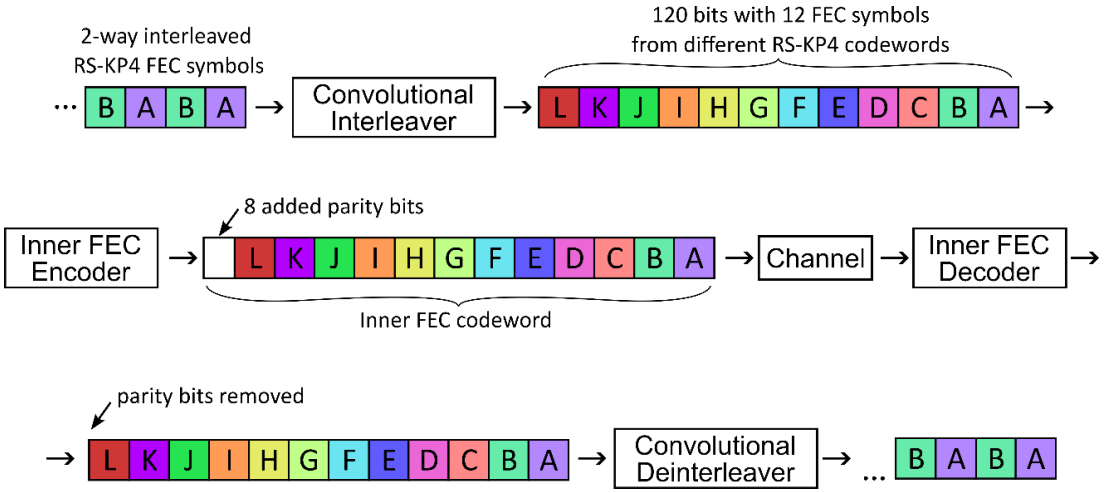


Figure 13. An example of convolutional interleaving and de-interleaving [2].

symbols should be distributed to different outer-FEC codewords. This is achieved with a combination of block and convolutional interleaving. Convolutional interleaving is a technique that permutes the order of input symbols at the cost of added latency [16]. It delays symbols by differing amounts, resulting in a shuffled transmission order. This shuffling is reversed on the receiving side by a convolutional de-interleaver, shown in Figure 13. A recent IEEE standard proposal defines the convolutional interleaver architecture to achieve an inner-FEC codeword with outer FEC symbols from 12 distinct KP4 codewords [17].

## 6.3 Soft-Decision Inner FEC Decoding

Soft-decision decoding can improve the inner FEC code's performance by using channel reliability information as well as the hard-decision PAM-4 symbols. This boost in performance comes at the cost of added complexity: a soft-output receiver is required to produce this reliability information; in addition, a more complex soft-decision inner-FEC decoder is needed.

Figure 14 shows the transceiver architecture for soft-decision inner-FEC decoding used on the FPGA platform. The receiver not only produces a sequence of hard-decision PAM-4 symbols $d_k \in \{0,1,2,3\}$, but also a series of reliability values $\alpha_k$. The reliability values should have the property that if $\alpha_k < \alpha_j$, then the probability $\Pr(d_k = b_k) < \Pr(d_j = b_j)$. In other words, symbols with higher $\alpha$ are more likely to be correct.

Each of the four possible output PAM-4 symbols can be rank-ordered from most to least likely. Typically, the third and fourth most likely PAM-4 symbols are negligible. In this case, the log-likelihood ratio of the first and second most likely symbols can be defined as the $\alpha$ that satisfies the inequality stated above. For example, if at timestep $k$ the most likely symbol is 0 and the second most likely symbol is 1, then

$$\alpha_k = \log\left(\frac{\Pr(b_k=0)}{\Pr(b_k=1)}\right) \quad (4)$$

By definition $\alpha \geq 0$, and $\alpha = 0$ only when the first and second most likely symbols are equally likely. With Gray-coded PAM-4 symbols, the second and first most likely

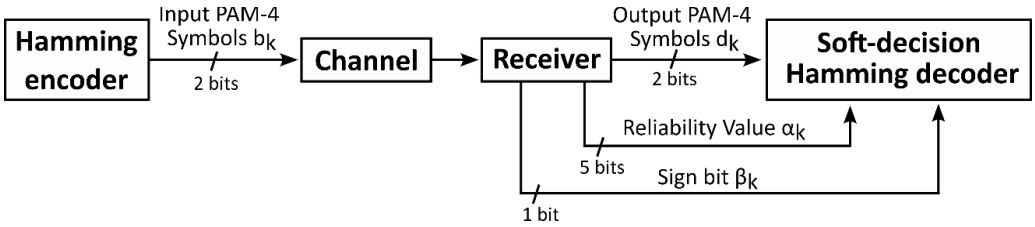


Figure 14. Transceiver architecture for soft-decision inner-FEC decoding [3].

symbols only differ by one bit, the location of this lower-reliability bit could be either in the most-significant bit (MSB) or least-significant bit (LSB) of the PAM-4 symbol. The location of the lower-reliability bit in each PAM-4 symbol is crucial for the soft-decision decoder to decide which bit to correct. The receiver passes on this information with an additional sign bit $\beta_k$. If $\beta_k = 0$, this signifies that the low-reliability bit is in the LSB of the PAM-4 symbol at time index $k$, and if $\beta_k = 1$, the low-reliability bit is in the MSB.

Chase presented an algorithm for soft-decision decoding of binary linear block codes in [20]. This algorithm is a popular choice for practical implementations of the inner-FEC decoder in 200 Gb/s system. A hard-decision decoder for a binary ($n$, $k$) code takes in a received sequence $y \in \mathbb{F}_2^{1\times n} = \{y_1, y_2, ..., y_n\}$, and finds the error sequence $z_m$ of minimum binary weight such that $y+z_m = \{y_1+z_{m1}, y_2+z_{m1}, ...y_n+z_{m1}\}$ is a valid codeword (here operator '+' represents the modulo-2 addition). Hard-decision binary decoders can only find error patterns of maximum weight $t$. For example, the decoder for the shortened Hamming code with $t = 1$ can only determine an error pattern of binary weight one. If a valid error pattern of weight one cannot be found, it experiences a decoding failure.

Similarly, a soft-decision decoder finds an error pattern $z_m$ of minimum analog weight such that $y+z_m$ is a valid codeword. The analog weight of an error pattern is defined as $\sum_{i=1}^{N} \alpha_i Z_{mi}$, which is the sum of the reliability values $\alpha$ in all positions for which the error pattern flips the received sequence of bits. The analog weight corresponds to the likelihood associated with that error pattern. Given a received sequence, an ideal soft-decision decoder finds the most likely codeword by enumerating all possible error patterns. Chase's algorithm for soft-decision decoding approaches the performance of an ideal soft-decision decoder.

Chase's algorithm finds a limited set of possible error patterns and picks the one with the lowest analog weight. An error pattern $z_T$ is found by perturbing $y$ by a test pattern $T$, and then applying hard-decision decoding on the codeword $y + T$. If the decoding is successful, the decoded codeword gives a new error pattern $z'$ with respect to $y + T$, hence $z_T = z' + T$.

Let $q$ and $w$ be integer parameters. The test patterns are generated by taking all combinations of $w$ or fewer 1s in the least reliable $q$ positions of the $Y$, including the all-0s test pattern. This gives a total of $\binom{q}{0}+\binom{q}{1}+\ldots+\binom{q}{w}$ test patterns. The Chase algorithm with parameters $q$ and $w$ can potentially extend the decoder's error-correcting capability from $t$ to $t + w$. By increasing $q$ and/or $w$, a larger number of test patterns are

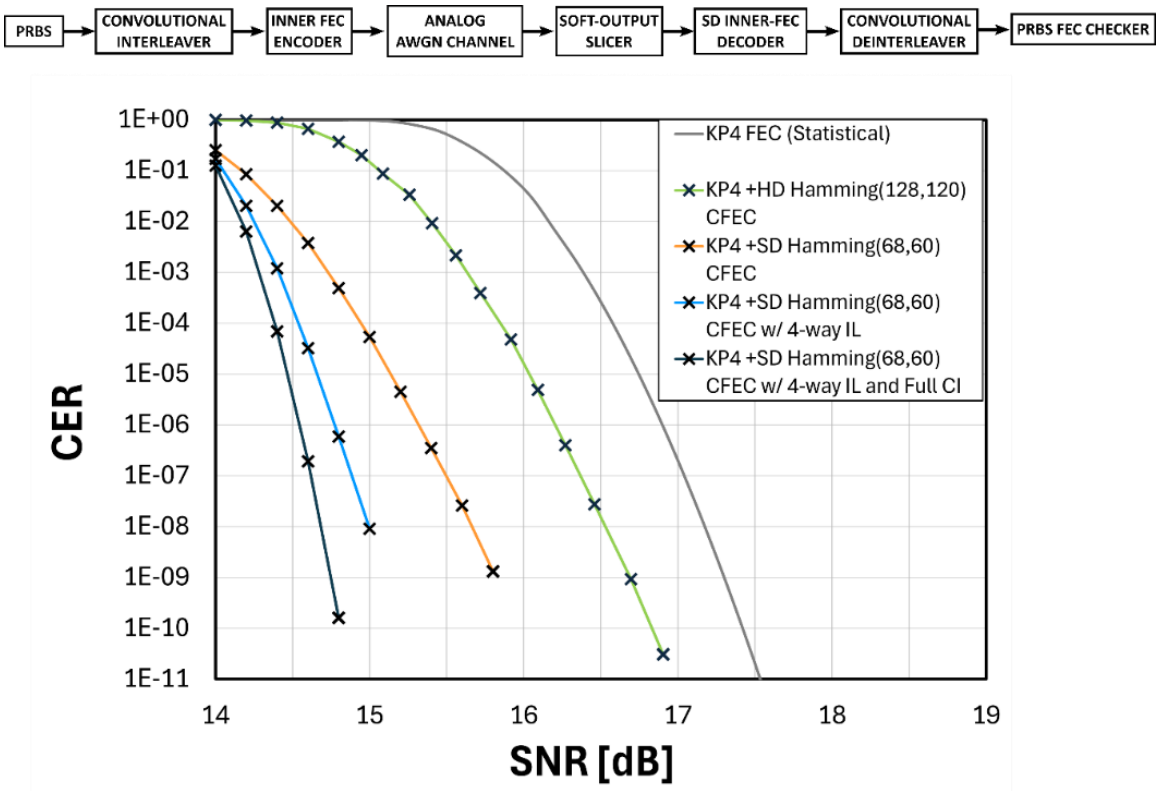


Figure 15. CER vs. SNR for the AWGN channel with soft-decision Hamming (68,60) inner code + KP4 outer code and various interleaving schemes.

considered in the algorithm, giving a better chance of finding the error pattern with the lowest analog weight. In this paper, $q$=6 and $w$=4 are chosen, giving 42 test patterns.

## 6.4 Soft Output Slicer

The platform includes a soft-output slicer for the AWGN channel. It not only determines hard-decided PAM-4 symbols $d_k$, but also the reliability values $\alpha_k$ and sign bits $b_k$ based on the distance of each received noisy sample $r_k$ from the target symbol levels. For the simulations shown in this paper $\alpha$ is represented with a five-bit value.

Figure 15 shows CER vs. SNR for the AWGN channel protected by the Hamming (68,60) inner FEC with soft-decision decoding and KP4 outer FEC. Simulation results with no interleaving, 4-way block interleaving, and 4-way block interleaving + full convolutional interleaving are shown. The same channel protected by a concatenated KP4 + hard-decision (HD) Hamming (128,120) FEC, and only a KP4 FEC, is included for comparison.

Soft-decision decoding provides significant boost in performance over the concatenated FEC with hard-decision decoding, and interleaving provides another large improvement. With 4-way block interleaving and full convolutional interleaving, the system shows a 2 dB SNR improvement over the same link protected only by the KP4 FEC.

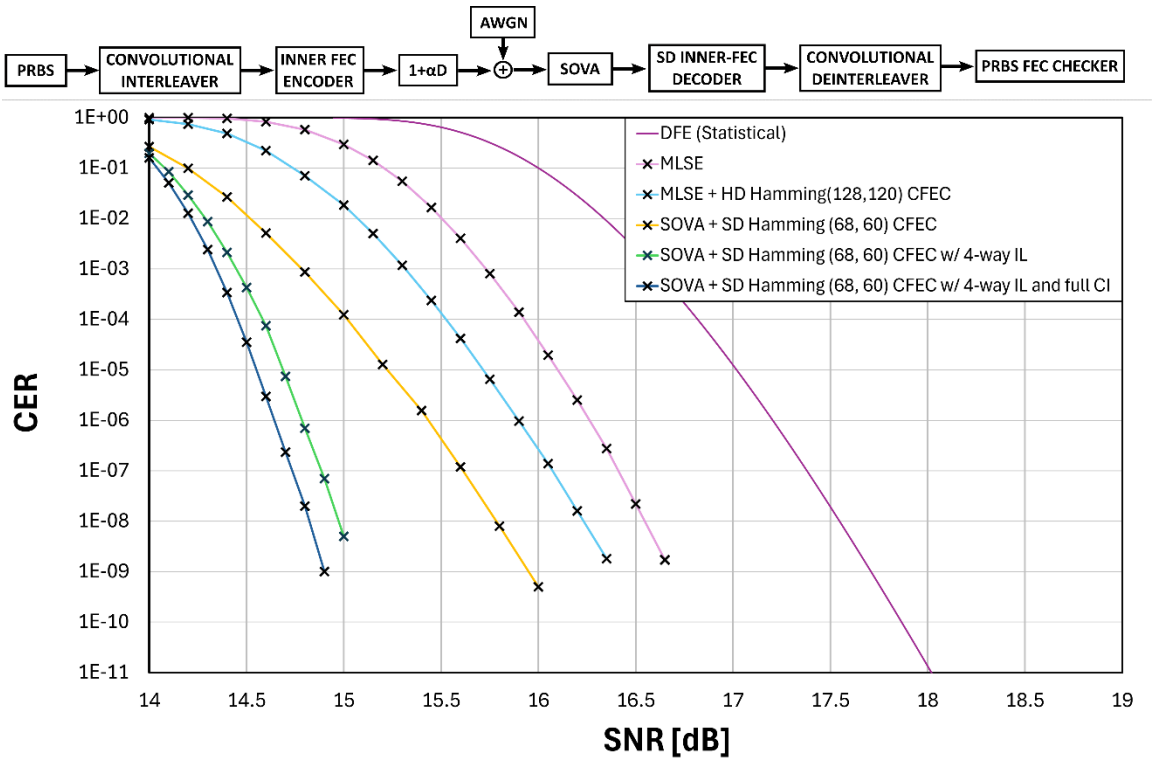


Figure 16. CER vs. SNR for 1+0.5*D* equalized with SOVA and protected by concatenated Hamming (68,60) + KP4 FEC.

## 6.5 Soft Output Viterbi Algorithm (SOVA)

In their 1989 paper, Hagenauer and Hoeher proposed the soft-output Viterbi algorithm (SOVA) [21]. SOVA is a modified Viterbi algorithm that not only outputs the maximum likelihood state sequence for a Markov process, but also a reliability value for each state (soft decisions). Their original paper only applies to equalizing links with PAM-2 modulation. Later, in 1999, Gong et al. proposed a generalized SOVA that can be applied to PAM-4 links [22]. This algorithm has become popular for 200 Gb/s optical receivers with soft-decision inner FEC decoding. The FPGA platform includes a module that performs SOVA on a 1+$\alpha D$ channel.

Figure 16 shows CER vs. SNR for the 1+0.5*D* equalized channel with SOVA and protected by the concatenated Hamming (68,60) + KP4 FEC codes. Simulation results are shown for this architecture with no interleaving, 4-way block interleaving, and with 4-way block interleaving and full convolutional interleaving. For comparison, several other architectures are shown to protect the same 1+0.5*D* channel.

The worst performance is from a DFE equalizer protected by the KP4 FEC only. A MLSD equalizer and KP4 FEC improves on the DFE by about 1 dB. The MLSD + concatenated hard-decision Hamming (128,120) + KP4 FEC improves further by about 0.4 dB. This is a relatively small improvement compared to the AWGN channel, which shows an improvement of around 0.7 dB when using the hard-decision inner FEC. This is because the correlated errors reduce the effectiveness of the Hamming (128,120) code

with $t = 1$. The use of SOVA and the soft-decision inner FEC code improves over the hard decision inner-FEC code by about 0.5 dB. Due to correlated PAM-4 symbol errors, this is also a smaller improvement than what is seen with the AWGN channel, which has an 0.8 dB improvement with the soft-decision inner FEC code over the hard-decision inner-FEC code. Finally, the interleaving schemes show another significant boost in performance by breaking up the correlated 120-bit-long inner-FEC codewords.

# 7. Case Study

Sections 4-6 introduced the custom Verilog modules that can be used to build a wireline system model for simulation on the FPGA platform. In this section, all these pieces are put together to demonstrate how the FPGA platform can be used in practice. A topic of ongoing discussion for 200 Gb/s multi-part links is the BER allocation between optical and electrical links, i.e., how high of a BER can be allowed in each of the three links while maintaining a sufficiently low post-FEC BER for the entire system? Correctly setting the BER allocation is essential to maintain interoperability for hosts and modules from different vendors.

This is particularly important for 200 Gb/s applications because circuit and interconnect designers are pushing the limits of available technology to achieve such a high data rate, and so they do not have the margin to set conservative BER requirements. Recent IEEE standard contributions have proposed target BERs [23], but until now there has not been a reliable method to validate the proposals rigorously.

This section presents a model of a realistic multi-part link on the FPGA platform. The analysis holds the electrical link's BER at a constant level and the optical link's BER is swept to find the level at which the overall system's CER is sufficiently low. This target CER level would ideally be $1.45\times10^{-11}$, the Ethernet Standard's specification. However, the hardware complexity of this model limits the FPGA's simulation speed to 1.2 Gb/s. At this speed, it would take weeks to run a simulation demonstrating a CER of $1.45\times10^{-11}$ with reasonable accuracy. An array of 20 parallel FPGAs could bring this time down to only one day. However, since only one FPGA is available for this research, this section assumes a target CER level of $1\times10^{-9}$ so that simulations may be performed in one day. Figure 17 shows the system-level model of a multi-part wireline system on the FPGA platform. This model represents the system detailed in Section 2.1, which does communication between data centers over a three-part electrical-optical-electrical link at 200 Gb/s per lane using the Ethernet protocol.

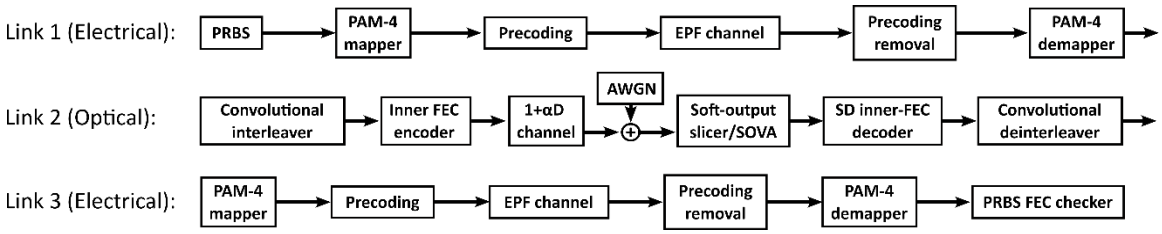


Figure 17. Case study: system-level architecture of a multi-part link with an AWGN optical link [3].

## 7.1 Model of Electrical Links

The electrical links are modelled with the error propagation factor model presented in Section 4.5.2. The initial error probability is $2.67\times10^{-5}$, and the error propagation factor is 0.75. Precoding is used, introducing a BER of $4\times10^{-5}$ by each of the electrical links. This model for host-to-module 200 Gb/s electrical links is taken from a recent IEEE Ethernet contribution [24].

## 7.2 Model of Optical Link

Two channel models are considered for the optical link: First, the analog AWGN channel presented in Section 5 with the soft-output slicer from Section 6.4. Next, the same system is analyzed with a 1+0.5$D$ channel, equalized with SOVA (presented in Section 6.5). These optical link models are based on other recent work analyzing error rates for 200 Gb/s links [23-24].

## 7.3 FEC Architectures

Three FEC architectures are compared in this analysis:

1. Full protection mode uses the concatenated soft-decision Hamming (68,60) and KP4 FEC codes. 4-way block interleaving and full convolutional interleaving are used to achieve the optimal inner-FEC codeword with 12 FEC symbols from distinct outer-FEC codewords.
2. Convolutional interleaver bypass mode disables the convolutional interleaver to reduce the system's latency with a cost to performance. This mode still uses the concatenated soft-decision Hamming (68,60) + KP4 FEC code with 4-way block interleaving.
3. Outer-FEC mode bypasses the inner FEC completely, leaving the three-part link protected only by the outer KP4 code. This mode significantly reduces the power and area requirements of the system by eliminating the inner FEC, at a large cost to performance. In this mode, the 1+0.5$D$ channel is equalized with MLSD, equivalent to SOVA without the soft reliability values.

Figure 18 shows the optical module's SNR on the x-axis and the corresponding overall KP4 CER for the entire system on the y-axis for the three different FEC architectures. Two channel responses 1+0$D$ (AWGN) and 1+0.5$D$ are used in this case study. For the 1+0$D$ channel, outer-FEC mode shows >2 dB degradation at the target CER level. Because of this, bypassing the inner FEC is only practical for a high-SNR optical link. Of the other two candidates, the convolutional interleaver bypass mode shows around 0.3 dB degradation compared to the architecture with full protection. For the 1+0.5$D$ channel, the outer-FEC mode shows a better performance compared to the 1+0$D$ channel. This is because MLSD benefits from a signal power that is amplified from the 1+0.5$D$ response. Despite this increased signal power, the optical channel now suffers from correlated PAM-4 symbol errors, reducing the effectiveness of the inner FEC code. As such, the other two modes show an overall degradation in performance by about 0.1 dB compared to 1+0$D$ channel. This study demonstrates how the FPGA platform may be used to develop BER allocations for the 200 Gb/s Ethernet standard.

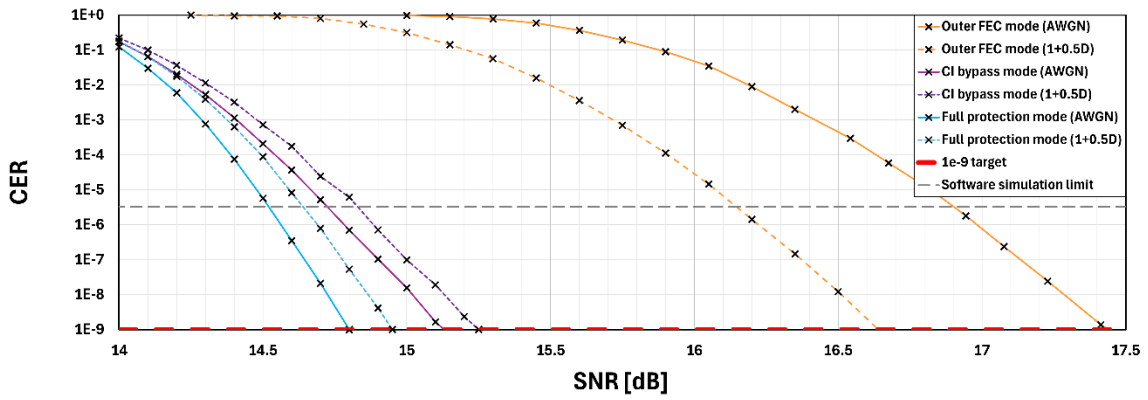


Figure 18. Overall link CER vs. Optical module's SNR of the multi-part link system in Figure 17 [3].

# 8. Conclusion

Tools to analyze the CER of 200 Gb/s wireline systems are needed to design reliable high-speed links and the standards to which they comply. These tools should be fast and flexible to estimate low CER levels for various channel types in a reasonable amount of time. Existing time-domain software platforms are prohibitively slow, and statistical methods lack the complexity required for realistic systems. This paper presented an FPGA-accelerated platform to address this demand. The platform allows for the modelling of complex wireline systems, including multi-part links with concatenated FEC, soft-decision decoding, block interleaving, convolutional interleaving, and precoding, with the flexibility to explore different channel types and FEC architectures. It provides a speed increase of more than 10,000 times over comparable time-domain software simulations [8-10], allowing for demonstration of the Ethernet standard's target CER of $1.45 \times 10^{-11}$ within one day of simulation time for a simple PAM-4 AWGN channel. In contrast, the time-domain software concatenated FEC model reported in [10] can only simulate down to $10^{-8}$ post-FEC BER level using 16 parallel CPU cores in one day. The prior FPGA work in [13] uses over-sampled time-domain signals, increasing channel-model accuracy but lowering simulation speed and it does not report BERs lower than $10^{-7}$ (~$10^{-5}$ CER). Although the platform can only demonstrate a CER of $10^{-9}$ for the full multi-part link model presented in Section 7, our proposed platform can easily be scaled up using a cluster of 18 or more FPGAs to validate CER levels $\leq 1.45\times10^{-11}$ within 24 hours. This significant speed improvement over existing simulation methods allows us to explore new and complex FEC architectures with high accuracy, which is the key novelty of our platform. This platform is specifically tailored for architectures relevant to 200 Gb/s Ethernet.

## Appendix A. Summary of FPGA platform speed

Table 1 shows a summary of the simulation speed achieved by our platform for the examples shown in this paper. The numbers reported in this table assume the maximum number of parallel cores are used subject to the timing constraints and the resources on the Xilinx VCU108 Board.

| Link Architecture Modelled on FPGA | Simulation Speed | Time to simulate $10^{13}$ bits |
|---|---|---|
| EPF channel model (Figure 7-8) | 30 Gb/s | 5 min |
| Analog AWGN channel, HD slicer | 9 Gb/s | 19 min |
| Analog 1+aD channel with AWGN and MLSD equalizer (Figure 11) | 6 Gb/s | 27 min |
| Analog AWGN channel with oft output slicer, SD inner FEC, convolutional interleaver (Figure 15) | 3 Gb/s | 56 min |
| 1+aD channel with AWGN, SOVA equalizer, SD inner FEC, convolutional interelaver (Figure 16) | 1.5 Gb/s | 111 min |
| Case study (Figure 18) | 1.2 Gb/s | 139 min |

Table 1. Summary of FPGA Simulation Speed

## References


[1] Y. Yin, The Role of Voltage Regulation in Power Integrity for Multi-Gbps Parallel I/O Interfaces, 2017.

[2] A. Ran and M. Brown, Error Probability Thoughts, IEEE standard 802.3dj, 2024. [Online]. Available: https://www.ieee802.org/3/dj/public/adhoc/electrical/24_0418/ran_3dj_elec_01a_240418.pdf.

[3] R. Barrie, M. Yang, H. Shakiba, and A. Chan Carusone, "An FPGA-Accelerated Platform for Post-FEC BER Analysis of 200 Gb/s Wireline Systems," IEEE Transactions on Circuits and Systems II.

[4] C. A. Belfiore and J. H. Park, "Decision Feedback Equalization," Proceedings of the IEEE, vol. 67, no. 8, pp. 1143–1156, 1979. doi: 10.1109/PROC.1979.11409.

[5] H. Shakiba, Error Propagation Analysis of MLSE, IEEE standard 802.3dj, 2023. [Online]. Available: https://www.ieee802.org/3/dj/public/adhoc/electrical/23_0420/shakiba_3dj_elec_02_230420.pdf.

[6] K. Wu, G. Liga, J. Riani, and A. Alvarado, "Low-Complexity Soft-Decision Detection for Combating DFE Burst Errors in IM/DD Links," Journal of Lightwave Technology, vol. 42, no. 5, pp. 1395–1408, 2024. doi: 10.1109/JLT.2023.3324602.

[7] "Draft Standard for Ethernet Amendment: Media Access Control Parameters for 1.6 Tb/s and Physical Layers and Management Parameters for 200 Gb/s, 400 Gb/s, 800 Gb/s, and 1.6 Tb/s Operation," IEEE Std 802.3dj-2024, 2024.

[8] M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev and A. C. Carusone, "Statistical BER Analysis of Wireline Links With Non-Binary Linear Block Codes Subject to DFE Error Propagation," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 1, pp. 284-297, Jan. 2020.

[9] M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev and A. Carusone, "A Statistical Modeling Approach for FEC-Encoded High-Speed Wireline Links", *DesignCon 2020*, Santa Clara, CA, 2020.

[10] R. Barrie, M. Yang, and A. C Carusone, "Statistical BER Analysis of Concatenated FEC in Multi-Part Links, " *Designcon 2023*, Santa Clara, CA, 2023.

[11] E. Lord, M. Devlin, N. Harold, and C. Sanderson, “Accelerating BER analysis using an FPGA based processing platform,” in MILCOM 2002. Proceedings, vol. 1, 2002, 218–223 vol.1. doi:10.1109/MILCOM.2002.1180443.
[12] W.Wang, Z. Long, W. Qian, et al., “Real-Time FPGA Investigation of Potential FEC Schemes for 800G-ZR/ZR+ Forward Error Correction,” Journal of Lightwave Technology, vol. 41, no. 3, pp. 926–933, 2023. doi: 10.1109/JLT.2022.3218957.
[13] D. Zou, K. Song, Z. Chen, C. Zhu, T. Wu, and Y. Xu, “FPGA-Based Configurable and Highly Flexible PAM4 SerDes Simulation System,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 9, pp. 1294–1307, 2023. doi: 10.1109/TVLSI.2023.3286803.
[14] C. J. Clopper and E. S. Pearson, “The Use of Confidence or Fiducal Limits Illustrated in the Case of the Binomial,” Biometrika, vol. 26, no. 4, pp. 404–413, Dec. 1934.
[15] S. Bhoja, W. Bliss, C. Chen, V. Parthasarathy, J. Wang, and Z. Wang, Precoding Proposal for PAM-4 Modulation, IEEE standard 802.3bj, 2011. [Online]. Available: https://www.ieee802.org/3/bj/public/sep11/parthasarathy_01_0911.pdf.
[16] G. C. Clark and J. B. Cain, Error-correction coding for digital communications. Plenum Press, 1988.
[17] A. Farhood, W. Bliss, S. Ramesh, and D. Cassan, Concatenated FEC baseline proposal for 200Gb/s per lane IM-DD Optical PMD, IEEE standard 802.3dj, 2023. [Online]. Available: https://grouper.ieee.org/groups/802/3/dj/public/23_01/23_0206/farhood_3dj_01a_230206.pdf.
[18] D.-U. Lee, J. Villasenor, W. Luk, and P. Leong, “A hardware Gaussian noise generator using the Box-Muller method and its error analysis,” IEEE Transactions on Computers, vol. 55, no. 6, pp. 659–671, 2006. doi: 10.1109/TC.2006.81.
[19] G. Forney, “The Viterbi Algorithm,” Proceedings of the IEEE, vol. 61, no. 3, pp. 268–278, 1973. doi: 10.1109/PROC.1973.9030.
[20] D. Chase, “Class of algorithms for decoding block codes with channel measurement information,” *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.
[21] J. Hagenauer and P. Hoeher, “A Viterbi Algorithm with Soft-Decision Outputs and its Applications,” in 1989 IEEE Global Telecommunications Conference and Exhibition’ Communications Technology for the 1990s and Beyond’, 1989, 1680–1686 vol.3. doi:10.1109/GLOCOM.1989.64230.
[22] L. Gong, W. Xiaofu, and Y. Xiaoxin, “On SOVA for nonbinary codes,” IEEE Communications Letters, vol. 3, no. 12, pp. 335–337, 1999. doi: 10.1109/4234.809527.
[23] A. Ran, Proposal for BER budget allocation for AUIs in Type 1 and Type 2 PHYs, IEEE standard 802.3dj, 2023. [Online]. Available: https://www.ieee802.org/3/dj/public/23_05/ran_3dj_01a_2305.pdf.
[24] H. Xiang, K. Huang, M. Dudek, and L. Patra, Low Latency Mode for Inner FEC, IEEE 802.3dj Task Force, 2023. [Online]. Available: https://www.ieee802.org/3/dj/public/24_01/he_3dj_01_2401.pdf.