Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Diseño Electrónico

TÍTULO DEL TRABAJO.
PAM4 Transmitter and Receiver Equalizers Optimization for High-Speed Serial Links

TESIS que para obtener el **GRADO** de
MAESTRÍA EN DISEÑO ELECTRÓNICO

Presenta: **ROBERTO JORGE RUIZ URBINA**

Director **FRANCISCO ELÍAS RANGEL PATIÑO**

Tlaquepaque, Jalisco. 4 de octubre de 2021.

Dedicated to my beloved family, my mother Elda, my father Roberto and my brother Eduardo for their support and encouragement.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Francisco Elias Rangel-Patiño for his guidance, and support. His passion for the research gave me the motivation to develop this thesis. I have benefited greatly from his wealth of knowledge, and I am extremely grateful to being his student.

Special thanks to Dr. Omar Humberto Longoria-Gandara, and Dr. Jose Ernesto Rayas-Sanchez from the ITESO Department of Electronics, Systems, and Informatics, and Mr. Edgar Andrei Vega-Ochoa from Intel Corporation, for their collaboration and suggestions to publish an IEEE paper as result of this thesis.

I would like to thank the financial support provided by Intel Corporation.

I would like to thank all my colleagues who supported me during this journey.

Finally, special thanks to my family, my mother Elda, my father Roberto and my brother Eduardo, for the endless support and encouragement

Summary

As the telecommunications markets evolves, the demand of faster data transfers and processing continue to increase. In order to confront this demand, the peripheral component interconnect express (PCIe) has been increasing the data rates from PCIe Gen 1(4 Gb/s) to PCIe Gen 5(32 Gb/s). This evolution has brought new challenges due to the high-speed interconnections effects which can cause data loss and intersymbol interference. Under these conditions the traditional non return to zero modulation (NRZ) scheme became a bottle neck due to bandwidth limitations in the high-speed interconnects. The pulse amplitude modulation 4-level (PAM4) scheme is been implemented in next generation of PCIe (PCIe6) doubling the data rate without increasing the channel bandwidth. However, while PAM4 solve the bandwidth problem it also brings new challenges in post silicon equalization. Tuning the transmitter (Tx) and receiver (Rx) across different interconnect channels can be a very time-consuming task due to multiple equalizers implemented in the serializer/deserializer (SerDes). Typical current industrial practices for SerDes equalizers tuning require massive lab measurements, since they are based on exhaustive enumeration methods, making the equalization process too lengthy and practically prohibitive under current silicon time-to-market commitments. In this master's dissertation a numerical method is proposed to optimize the transmitter and receiver equalizers of a PCIe6 link.

The experimental results, tested in a MATLAB simulation environment, demonstrate the effectiveness of the proposed approach by delivering optimal PAM4 eye diagrams margins while significantly reducing the jitter.

Contents

Introduction

Nowadays, peripheral component interconnect express (PCIe) is one of the most complex standard computer interfaces. It is based on a high-speed serial packet-switching communication technology that evolves with the new computer industrial trends and demands. PCIe constitutes the primary computer interface for connecting a host central processing unit (CPU) to input/output (I/O) peripheral devices, and virtually operates in all modern computer systems.

In most of these systems, the PCIe bus co-exists with one or more legacy PCI buses, for backward compatibility with the large body of legacy PCI peripherals. Since 2013, PCIe has replaced the accelerated graphics port as the default interface for graphics cards on modern systems.

Being an open industry standard, PCIe has succeeded as a global I/O interconnect supported by a robust compliance program to ensure a unified interoperability between devices from different companies.

The PCIe bandwidth has been scaled by means of multiple lanes (×1, ×2, ×4, ×8, ×16, and ×32) and the interconnection rate has been doubled every generation with full backward compatibility every 3 years, increasing from 2.5 gigabyte per second (Gb/s) on the first generation to 32 Gb/s for the current fifth generation.

Even though there is a constant increase in PCIe bandwidth, new applications still require data rates to increase. However, as the data rate increases beyond 32.0 Gb/s, the bandwidth becomes the bottleneck of high-speed wireline transceiver, which is severely influenced by the channel and package loss by using the conventional non-return-zero (NRZ) signaling method.

In order to overcome this problem, the next generation of PCIe will adopt the pulse amplitude modulation 4-level (PAM4) signaling, which has been widely used by networking standards as they moved onto data rates of 56 Gb/s and beyond. PAM4 encodes two bits into one symbol, which achieves the same data rate using half of the bandwidth compared to the NRZ signaling. In this sense, this transmission scheme has a spectral efficiency of 2 bits/symbol/Hz

While PAM4 solves the bandwidth issue in high-speed communication channels, it brings new challenges for the link-path analysis having 4 levels and three eye-diagrams, as opposed to one eye-diagram of NRZ. PAM4 is also more susceptible to errors due to various noise sources caused by reduced voltage (and timing) ranges; i. e., the signal to noise ratio is reduced and the eye-diagram is more closed than the NRZ eye-diagram. This results in a higher bit error rate (BER) performance several orders of magnitude higher than the standard 10^{-12} BER for the five previous PCIe generations; and finally, it introduces new challenges in slicers, transition jitter and equalizers.

In this context, equalization (EQ) plays an important role even with PAM4 signaling. Transmitter (Tx) and receiver (Rx) equalization schemes such Tx side de-emphasis and pre-emphasis; and Rx side continuous time linear equalization (CTLE) and decision feedback equalization (DFE) are widely used in high-speed serial links to open the eye, and they continue to be used for PAM4.

PCIe6 specification defines the requirements to perform on-chip EQ at the Tx and at the Rx to mitigate undesired effects and minimize the BER. For the Tx EQ, the signal can be reshaped before the signal is transmitted in order to overcome the distortion and the impairments introduced by the communication channel. At the Rx, the received signal can be reconditioned to improve the signal quality.

PCIe6 specification defines an adaptive mechanism for EQ to determine the optimum value of the Tx and Rx EQ coefficients within a fixed time limit. The standard method to find out the best subset of coefficients consists of using maps of EQ coefficients, which are obtained by measuring the eye height, and eye width of the received signal. These maps show how the Rx performs at different locations of the coefficient space.

A typical PCIe system may have hundreds of combinations of EQ coefficients, and some of these combinations will produce better EQ results than others. Along with all the new challenges imposed by PAM4 and the additional manufacturability considerations, testing every coefficients combination using an exhaustive methodology to find the best one is impractical, and then optimization algorithms are required for choosing the right coefficients values.

This thesis presents an efficient optimization methodology to determine the optimal subset of coefficients for the Tx and Rx in a PCIe Gen6 equalization process during post-silicon validation. While there are not still silicon samples with PCIe Gen6, the evaluation of the proposed method is performed by using the MATLAB SerDes Toolbox. This thesis is organized as follows.

In Chapter 1, the evolution of the PCIe protocol is discussed, it covers an overview of the multiple PCIe generations developed through the years to satisfy the bandwidth demand of segments like data centers, audio/video streaming, gaming, etc., and how this evolution came with multiple challenges leading to the adoption of PAM4 for PCIe6.

Chapter 2 provides an overview of PAM4, the gray coding, the eye diagram anatomy for PAM4, and the challenges and opportunities by using this new modulation scheme.

Chapter 3 covers the PCIe equalization architecture for the Tx and Rx equalizers. It describes the implementation of the Feed Forward Equalizer (FFE) filter at the Tx, and the operation of the CTLE and DFE at the Rx to recover the signal from the Tx. This chapter also provide an overview of the adaptative equalization process of PCIe to find the best EQ for the Tx and Rx equalizers.

In Chapter 4, the new optimization methodology to find out the optimal subset of coefficients for the Tx and Rx in a PCIe6 equalization process is presented. The procedure implies defining an objective function based on a new FOM as required for PAM4 and then applying a direct numerical optimization based on Nelder-Mead. The simulation environment and results are also presented in this chapter.

In the General Conclusions, the most relevant remarks about this master dissertations are summarized. Finally, appendix A shows the MATLAB code developed to simulate and optimize the PCIe6 link.

1. PCI Express Evolution

Peripheral component interconnect express, officially abbreviated as PCIe is a high-speed serial computer expansion bus standard.

PCIe virtually operates in all modern computer systems from enterprise data servers, data storage, telecommunications, desktop, laptops, entertainment, imaging systems to industrial applications as a motherboard-level interconnect, a passive backplane interconnect and as an expansion card interface for add-in boards; and recently being adopted for automotive Advanced Driver Assistance Systems (ADAS) [5].

Nowadays, PCIe is one of the most complex standard computer interfaces. It is based on a high-speed serial packet-switching communication technology that evolves with the new computer industrial trends and demands [1]. PCIe constitutes the primary interface for connecting a CPU to I/O peripheral devices through a star-topology architecture that has strong correlation with the modern switched Ethernet fabric [2].

Being an open industry standard, PCIe has succeeded as a global I/O interconnect supported by a robust compliance program to ensure a unified interoperability between devices from different companies [6].

In this chapter, a description of PCIe is provided and a brief historic evolution of this interface is presented.

1.1. PCI Express

Figure 1-1 illustrates an example of a PCIe switched architecture composed of a root complex (RC) that connects the CPU, the memory subsystem, and the graphics controller to the I/O devices, multiple endpoints (I/O devices), switch components, and a PCIe to PCI bridge, all interconnected via PCIe links. As illustrated in Fig. 1-1, an RC may support one or more PCIe ports. Each interface defines a separate hierarchy domain that may be composed of a single endpoint or a sub-hierarchy containing one or more switch components and endpoints [1].

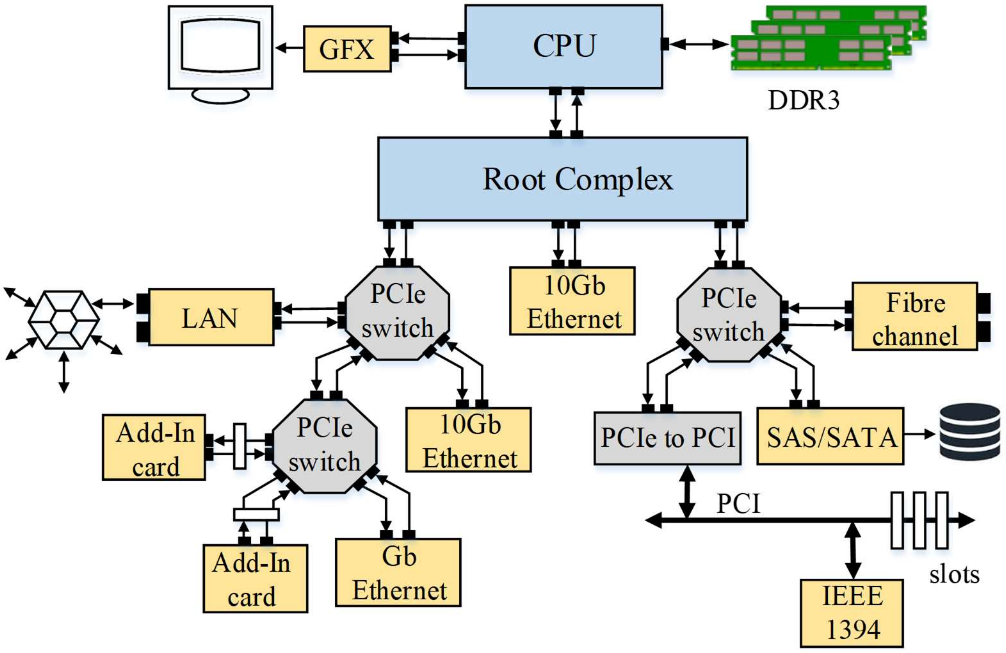


Fig. 1-1 PCIe switched architecture. Figure taken from [1].

PCIe uses a bidirectional connection, and it is capable of sending and receiving information at the same time. The model used is referred to as dual-simplex connection because each interface has a simplex transmit path and a simplex receive path, as shown in Fig. 1-2. The term used to describe this path between the devices is called link, and it is made up of one or more transmit and receive pairs. One such pair is called a lane, and the specification allows a link to be made up 1, 2, 4, 8, 12, or 32 lanes. The number of lanes is called the link-width and is represented as ×1, ×2, ×4, ×8, ×12, ×16, and ×32 [3]. The link can dynamically down-configure itself to use fewer lanes, providing a failure tolerance in case bad or unreliable lanes are present.

1.2. PCI Express Evolution

The peripherical component interface (PCI) bus was developed in the early 1990’s to address the limitations of the peripheral buses being used in personal computers (PC) at the time. The standard at the time was IBM’s advanced technology (AT) bus, referred also as industry standard architecture (ISA) bus. ISA was sufficient for the 286 16-bit machines, but additional bandwidth was needed for the newer 32-bit machines. The ISA big connector with high pin count was another problem. Then, several alternative bus designs were proposed, such IBM’s micro-

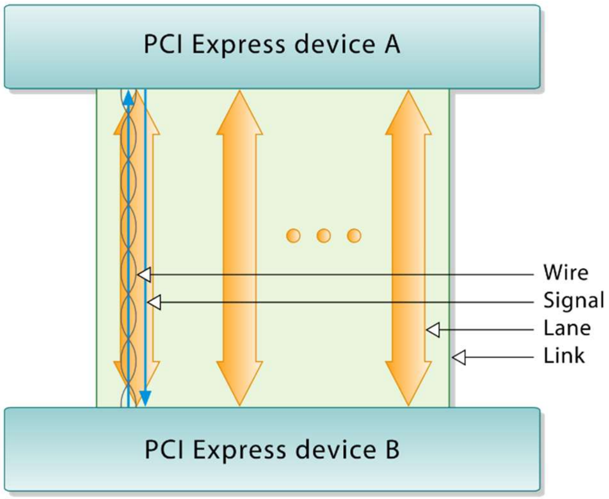


Fig. 1-2 PCI Express link between two devices consists of one or more lanes, which are dual simplex channels using two differential signaling pairs.

channel architecture (MCA), the extended ISA (EISA) bus, and the video electronics standards association (VESA) bus [3]. However, all these designs had several disadvantages preventing extensive industry acceptance. Eventually, PCI was developed as an open standard by a consortium of main PC developers who formed a group called PCI special interest group (PICSIG). The PCI features such open design, high speed, and software visibility and control, allowed PCI to overcome the issues shown by ISA and other buses. Therefore, PCI rapidly became the standard peripheral bus in the PC market.

Few years later, PCI-eXtended (PCI-X) was developed as a logical extension of the PCI architecture and improved the performance of the bus. Later, the PCI-X 2.0 revision added even higher speeds, achieving a raw data rate of up to 4 GB/s. Since PCI-X kept hardware compatibility with PCI, it remained a parallel bus and inherited the problems associated with that model. The speed limitation of PCI-X along with a high pin count, motivated the transition away from parallel bus model to the new serial bus model [3]. These earlier bus definitions are shown in Table 1-1.

PCIe represents a major shift from the parallel bus model of its predecessors. As a serial bus, it has more in common with earlier serial designs like InfiniBand or fibre-channel, but it remains fully backward compatible with PCI in software.

The PCIe bandwidth has been scaled by means of multiple lanes and the interconnection rate has been doubled every generation with full backward compatibility every 3 years, increasing

1. PCI EXPRESS EVOLUTION

TABLE 1-1.

COMPARISON OF BUS FREQUENCY, BANDWIDTH AND NUMBER OF SLOTS. FROM [3]

Bus Type	Clock Frequency	Peak Bandwidth 32-bit – 64-bit bus	Number of Card Slots per Bus
PCI	33 MHz	133 – 266 MB/s	4 - 5
PCI	66 MHz	266 – 533 MB/s	1 - 2
PCI-X 1.0	66 MHz	266 – 533 MB/s	4
PCI-X 1.0	133 MHz	533 – 1066 MB/s	1 - 2
PCI-X 2.0 (DDR)	133 MHz	1066 – 2132 MB/s	1 (point to point bus)
PCI-X 2.0 (QDR)	133 MHz	2132 – 4262 MB/s	1 (point to point bus)

from 2.5 Gb/s on the first generation (PCIe1 - 2003), to 5 Gb/s (PCIe2 - 2007), 8 Gb/s (PCIe3 - 2010), 16 Gb/s (PCIe4 - 2017) and 32 Gb/s (PCIe5 - 2019) [6] as shown in Fig. 1-3 and table 1-2.

Even though there is a constant increase in PCIe bandwidth, new applications such as artificial intelligence, machine learning, gaming, visual computing, storage, graphics accelerators, high-end networking, coherent interconnects, internet of things (IoT) and memory expanders still requires a data rate increase [6].

However, as the data rate increases beyond 32.0 Gb/s, the bandwidth becomes the bottleneck of high-speed wireline transceiver which is severely influenced by the channel and package loss by using the conventional non-return-zero (NRZ) signaling method [7]. Exotic printed circuit board (PCB) materials may compensate for these deficiencies, but at a cost that few (e.g., military, aerospace) are willing to pay.

In order to overcome this problem, the next generation of PCIe (PCIe Gen6) will adopt the pulse amplitude modulation 4-level (PAM4) signaling, which has been widely used by networking

TABLE 1-2.

EVOLUTION OF PCIE SPECIFICATIONS. FROM [4]

PCIe Specification	Data Rate(Gb/s) (Encoding)	Year
1.0	2.5 (8b/10b)	2003
2.0	5.0 (8b/10b)	2007
3.0	8.0 (128b/130b)	2010
4.0	16.0 (128b/130b)	2017
5.0	32.0 (128b/130b)	2019
6.0 (in progress)	64.0 (PAM-4)	2021

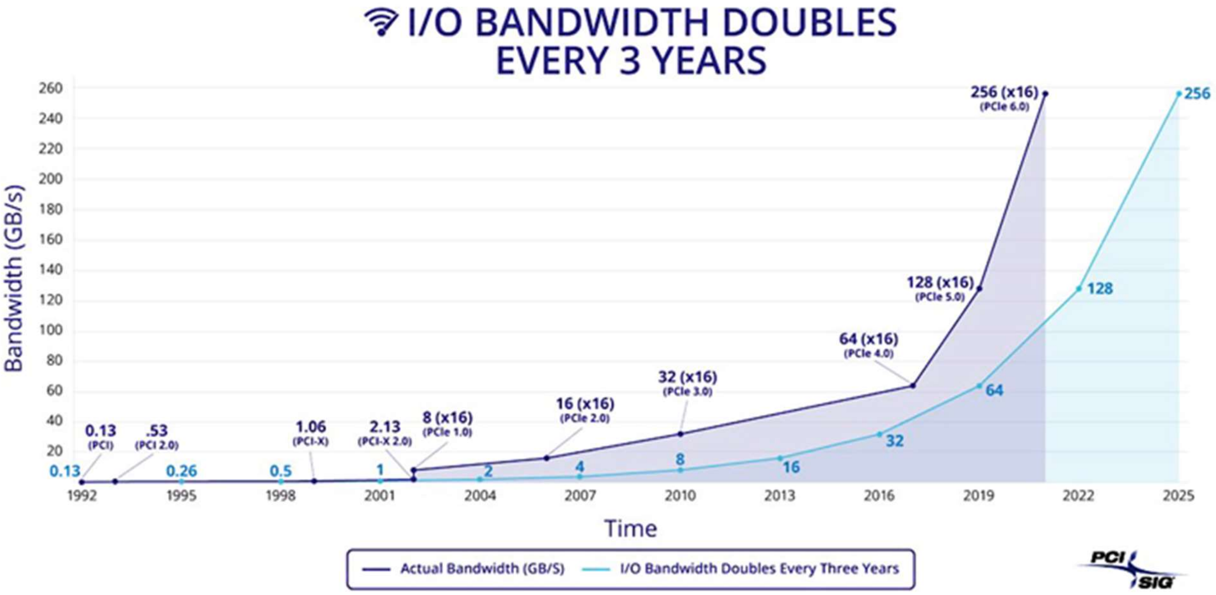


Fig. 1-3 The PCI Express roadmap, demonstrating the doubling per-pin bandwidth every generation. Figure taken from [6].

standards as they moved onto data rates of 56 Gb/s and beyond. PAM4 encodes two bits into one symbol, which achieves the same data rate using half of the bandwidth compared to the NRZ signaling. In this sense, this transmission scheme has a spectral efficiency of 2 bits/symbol/Hz.

1.3. Conclusions

Due to the emergence of new technologies such as data centers, audio/video streaming, gaming, and artificial intelligence; the demand of transmitting big amounts of data has increased. PCIe has been forced to evolve through the years to satisfy those demands, and the tendency of the communication markets is to continue demanding even higher data transfers. This situation is had led PCIe to adopt a new modulation scheme like PAM4 to satisfy the growing demand of bandwidth.

2. Four-level Pulse Amplitude Modulation

The constant development of telecommunications to satisfy the data demand in cloud computing, 5G networking, and media sharing has led to a higher bandwidth requirement on communication networks. Due to that demand, it is expected the data rate per lane for multiple protocols reach over 400 Gb/s very soon. However, the high-speed data transmission technology has reached an inflection point. The conventional binary NRZ signal modulation can no longer maintain the signal integrity quality required for reliable data transfer due the limitations of the channel bandwidth [7].

2.1. Four-level Pulse Amplitude Modulation

In order to overcome the bandwidth limitations in the high-speed data transmissions, the PAM4 signaling has been proposed to replace the NRZ modulation and become the main signaling scheme.

Compared to the traditional NRZ modulation with two output levels and two kinds of transition edges, PAM4 has 4 output levels and 12 types of transition edges as shown in Fig. 2-1. Instead of one eye in the NRZ coding scheme, there are three eyes in PAM4 because of the four voltage levels. The naming conventions to represent these four voltage levels are: -3, -1, 1, 3 or -1, -1/3, 1/3, 1 or 0, 1, 2, 3 (see Fig. 2-1).

The four output levels directly compress the output swing to one third of that in the NRZ modulation due to the signal to noise ratio (SNR), while the edge transitions between adjacent and nonadjacent output levels cause additional switching jitter. This switching jitter could become a critical issue as the data rate increases because the edge-transition time can cover a large portion of the short unit interval (UI) [8].

2.2. Gray coding

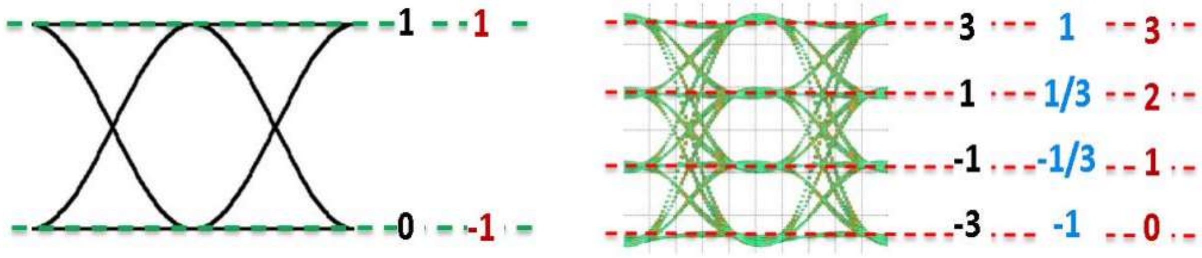


Fig. 2-1 NRZ and PAM4 comparison. Figure taken from [12].

The Gray code represents numbers using a binary encoding scheme that groups a sequence of bits so that only one bit in the group changes from the number before and after.

For PAM4 encoding, the signal has four voltage levels, which encodes two bits per voltage level, as shown in Fig. 2-2. PAM4 uses Gray coding which combines the most significant bit (MSB) and least significant bit (LSB) pairs in a data stream into one of the four voltage levels. By encoding two bits into one symbol, PAM4 achieves the same data rate using half of the bandwidth as compared to the NRZ signaling [9] (see Fig. 2-2). In this sense, this transmission scheme has a spectral efficiency of 2 bits/symbol/Hz.

2.3. PAM4 Eye Diagram Anatomy

In PAM-4, different considerations must be taken to measure the eye height and eye width, the signal loss for PAM-4 is 20*log10(1/3) = 9.5dB due to three voltages levels in the eye. Compared to NRZ's two voltage levels, PAM-4 has four voltage levels that result in 12 distinct signal transitions, (six rise & six fall times) creating three district eye openings.

The Fig. 2-3 describes how the symmetry affects the eye height and eye width measurements; the largest eye width does not correspond to the largest eye height. When using NRZ, the eye height and width are measured from the biggest opening of an eye. However, this is not the case for PAM4's eye height and eye width. For the PAM4 eye, one needs to locate first the midpoint (T_{mid}) of the maximum horizontal eye opening [10] as shown in Fig. 2-3.

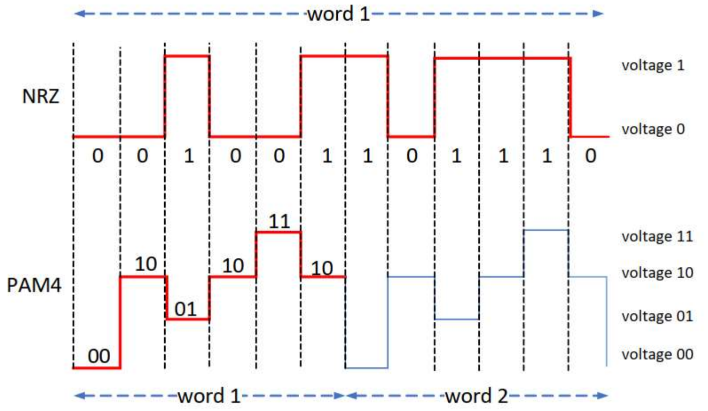


Fig. 2-2 Comparison of NRZ and PAM4 encoding. Figure taken from [9].

Once the T_{mid} is found, a vertical line can be drawn in such a way that intersects with the three eyes' 10^{-6} contour ring. EH6 (Eye Height BER of 10^{-6}) is the vertical distance between two intersection points on the 10^{-6} contour ring in an eye. EH6 represents the eye height at a BER of 10^{-6} and it is not necessarily the maximum eye height.

EW6 (Eye Width BER of 10^{-6}) represents the eye width at a BER of 10^{-6}. Taking the upper eye for example, once the half point of the eye height (EH6 upp)/2 is found, a horizontal line can be drawn in such a way that intersects with the 10^{-6} contour ring. The EW6 of the upper eye is the horizontal distance between two intersection points on the 10^{-6} contour ring in the eye. The lower eye's EW6 is measured in the same way. From Fig. 2-3 it can be noticed that the EW6 of each eye is not the widest opening. The asymmetry of the upper and lower eye causes the widest portion of the eye to be off-center. Compared with the widest portion, EW6 is considerably reduced.

2.4. PAM 4 challenges and opportunities

Increased demands for a connected world with instant data access continues to drive network transmission innovation. Development of 100 Gb/s data transmission is currently in production and will continue to evolve. Achieving 400 Gb/s data transfers represent a huge technological improvement. Technological advances towards achieving faster data transfers speed represents multiple possibilities, and challenges for PAM4.

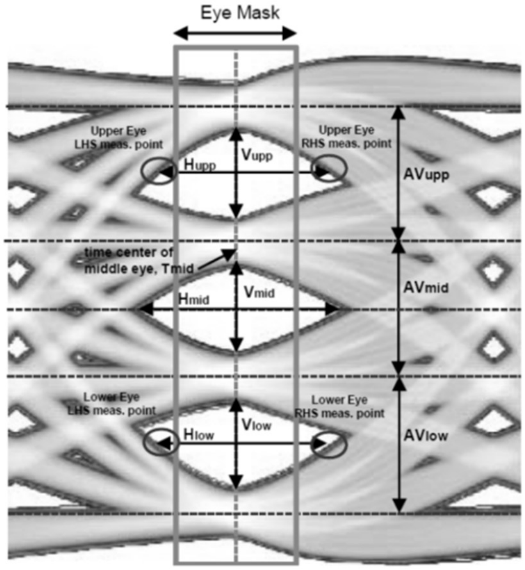


Fig. 2-3 PAM4 Horizontal Eye mask Sample. Figure taken from [3].

The familiar challenging concepts for NRZ also apply to PAM4 to include totally closed eyes, shorter unit intervals (UI), tighter jitter requirements, and the mandatory use for forward error correction (FEC). The closed eye issue causes triggering difficulties as with 100 Gb/s, but with more channel loss at 400 Gb/s and requires enhanced receiver equalization such as CTLE and DFE to correct. Standards are requiring increased receiver sensitivity (down to 50 mV). Jitter budgets are even tighter for 400 Gb/s at 17ps UI and may be below intrinsic jitter of test equipment. FEC, a technique used for controlling errors in data transmission over unreliable or noisy communication channels, becomes a greater challenge with increased noise at the faster data rate. For the most part, channel loss and reflections (noise) are expected to be the biggest PAM4 technological challenge as it continues growing [11].

Since the symbol rate with PAM4 is half that of NRZ, the signal suffers less from channel loss. But other effects like inter-symbol interference (ISI), reflections and crosstalk are higher in PAM4 [12].

Equalization is one of the most important components of high-speed design. Its goal is to recover a signal corrupted due to channel loss, inter-symbol interference, jitter, noise and crosstalk. Typical channel architecture includes transmitter side pre-emphasis, as well as receiver side Feed-Forward Equalizer and DFE.

For proper implementation of the equalizers, optimization algorithms are necessary for choosing correct tap coefficients. Finding most efficient tap coefficients could decrease the number of necessary DFE taps, therefore reducing cost of the equalizer [13].

The clock and data recovery (CDR) circuit is the core of the wireline transceiver. Many challenges exist in the PAM4 CDR design with 9 dB lower SNR and four voltage levels, especially at the data rate over 50 Gb/s. The noise sensitive CDR needs higher input voltage swing and lower noise, which demands high performance analog front-end [7].

Some challenges in the design will include the CDR due to the transitions and ISI. Time transitions in PAM4 can create horizontal eyes closed due to the switching jitter because of the dependency of the rise and fall times transitions in the signal.

The DFE will also play an important role due to the ability to correct the logical decision thresholds on the waveforms. The PAM4 signal has 1/3 the amplitude of that of a similar NRZ signal (SNR loss of ~9.5 dB) due to level spacing and is more susceptible to noise. However, it is possible that the lower PAM4 insertion loss compensates for the 9.5 dB loss in SNR due to reduced signal amplitude in PAM4 signaling.

2.5. Conclusion

While PAM4 solves the bandwidth issue in high-speed communication channels, it brings new challenges for the physical channel analysis. PAM4 has 4 levels and three eye-diagrams, as opposed to one eye-diagram of NRZ. PAM4 is also more susceptible to errors due to various noise sources caused by reduced voltage (and timing) ranges. This results in a higher bit error rate (BER) performance several orders of magnitude higher than the standard 10^{-12} BER of the previous PCIe generations. It also introduces new challenges in slicers, transition jitter, and equalizers. In effect, EQ plays a critical role even with PAM4 signaling.

3. PCI Express Equalization

PCIe6 specification defines an adaptative mechanism for the EQ to determine the optimum value of the Tx and Rx EQ coefficients within a fixed time limit. A typical PCIe system may have hundreds of combinations of EQ coefficients and then testing every coefficients combination using exhaustive enumeration method becomes prohibitive. In order to reduce the selection time, the current post-silicon method for Gen3-Gen5 consist of finding a subset of coefficients during post-silicon validation and then program it into the BIOS. The method consists of using maps of EQ coefficients, which are obtained by measuring the eye diagram characteristics as figure of merit (FOM). The method consists of finding the set of coefficients that qualify the FOM as near optimal [9].

Tx and Rx equalization schemes, such as Tx de-emphasis and pre-emphasis, Rx CTLE and DFE, are widely used in high-speed serial links to open the eye diagram [14],[15], and they continue to be used for PAM4. PCIe Gen6 specification defines the requirements to perform on-chip EQ at the Tx and at the Rx to mitigate undesired effects and minimize the BER.

3.1. Transmitter and receiver equalizers

PCIe6 implements a FFE 4-tap FIR filter. Fig. 3-1 shows a block diagram of a FIR filter, where C_{m2}, C_{m1}, C_0, and C_{p} represent the four filter taps coefficients. The precursors (C_{m2}, C_{m1}) and post-cursor (C_{p}) coefficients refer to whether the FFE filter taps work on an advanced or delayed signal with respect to time [16].

Through the FFE filter, the serial data signal is delayed by several flip-flops which implement the filter taps. Three consecutive received pulses ($v\mathrm{n}_{\mathrm{m2}}$, v_{nm1}, v_{n}, v_{np}) are multiplied with the four different filter tap coefficients, and the results are summed and driven to the serial data output.

The filter response can be changed by controlling the tap coefficients values. Therefore, the output signal (v_{out}) of the FIR filter is given by

3. PCI Express Equalization

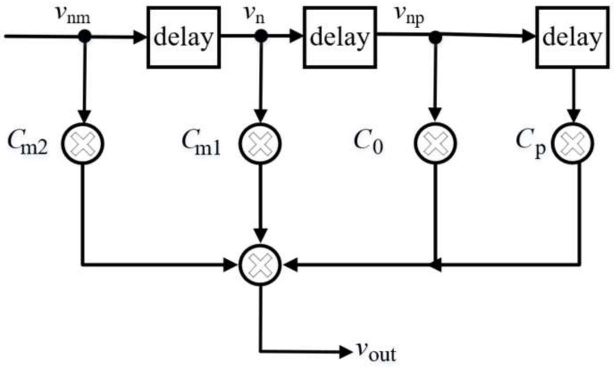


Fig. 3-1 FFE 4-tap FIR for PCIe Gen 6.

$$v_{out} = v_{nm2}C_{m2} + v_{nm1}C_{m1} + v_nC_0 + v_{np}C_p \tag{3-1}$$

The pre-emphasis/de-emphasis is implemented at the Tx driver by pre-conditioning the signal before transmitting it through the channel. Through this mechanism, the high-frequency content of the transmitted signal is amplified (pre-emphasis) or the low-frequency content of the signal is decreased (de-emphasis).

The EQ topology at the Rx can be a combination of a CTLE that works independently of the clock recovery circuit, and a DFE as shown in Fig. 3-2. The CTLE is a continuous-time circuit with high-frequency gain boosting, whose transfer function can compensate the channel response. The topology is an RC network with a parallel source-coupled pair, as shown in Fig. 3-3. The differential-pair emitter resistor attenuates the low frequency signals while the capacitor allows the high-frequency signal content, thus resulting in high boosting gain improving as shown in Fig. 3-4.

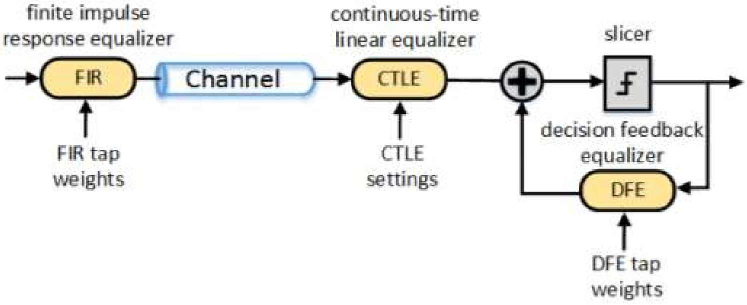

finite impulse response equalizer
FIR
Channel
continuous-time linear equalizer
CTLE
slicer
FIR tap weights
CTLE settings
decision feedback equalizer
DFE
DFE tap weights

Fig. 3-2 PCIe EQ topology. Figure taken from [17].

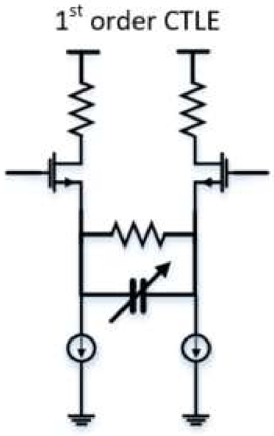


Fig. 3-3 CTLE RC circuit. Figure taken from [17].

Similar to Tx pre-emphasis, the CTLE addresses pre-cursor and post-cursor ISI but in continuous time instead being limited to a pre-set number of Tx taps. This simple topology has a very low power consumption rate, and many equalizer stages can be added not only to increase the order of the resulting equalizer but also to increase the maximum boost achieved in a given frequency interval.

The DFE consists of a feedforward filter which determines the value of the current transmitted symbol and passing it through the feedback filter, then the resulted ISI is removed from the upcoming symbols as shown in Fig. 3-5.

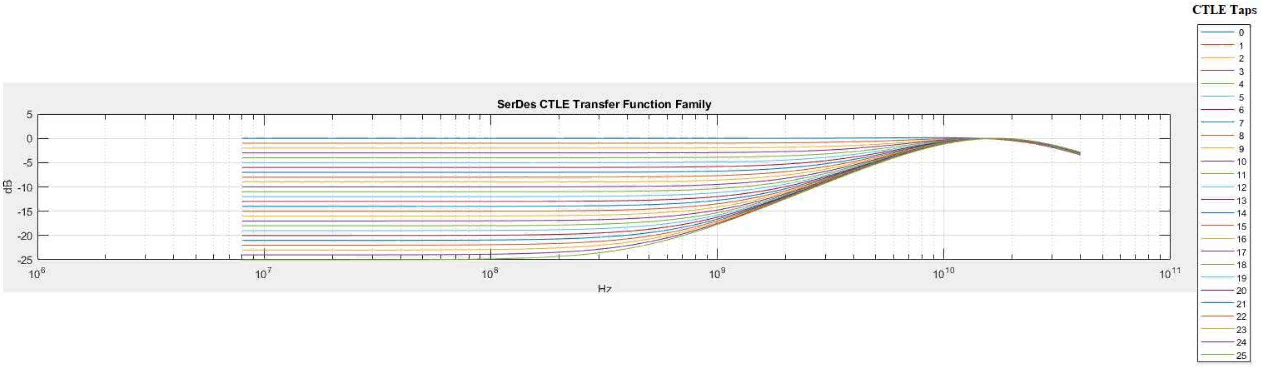


Fig. 3-4 Twenty-five taps CTLE Frequency response.

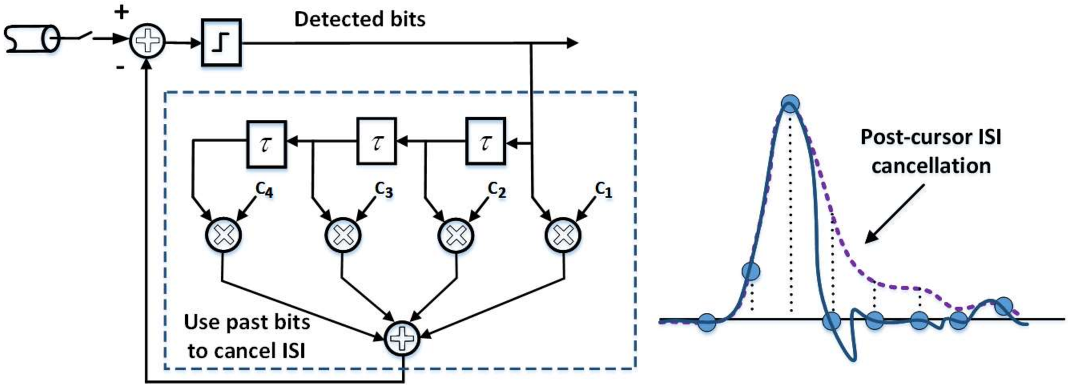


Fig. 3-5 Decision Feedback Equalizer. Figure taken from [17].

3.2. Adaptative equalization

PCIe6 specification establishes some predefined set of values for the four Tx coefficients, which are referred to as presets, and then are adaptively changed during the link training and equalization procedure, in which both downstream port and upstream port devices negotiate each to other the Tx EQ values to guarantee a BER less than 10^{-6} as shown in Fig. 3-6. Since the Tx does not know the channel parameters, the Tx EQ coefficients are computed at the upstream port by the coefficient adaptation algorithm in the medium access control (MAC) layer using the received signal. Then, these coefficients are communicated to the downstream port by using the PCIe protocol. The Tx at the downstream port then applies the received coefficients setting to the Tx EQ circuitry. The Rx drives two types of quality feedback by measuring the eye opening or evaluating eye edge ISI [17]. This process of computing the coefficients, communicating them to the Tx, and checking the signal quality can be repeated multiple times until the required BER is achieved.

The equalization procedure consists in four phases:

- Phase 0: Downstream port transmits the Tx preset values to the upstream port running either at previous PCIe generations.
- Phase 1: Downstream port and upstream port handshake and ensure a BER < 10^{-4} before the link is ready to move to the next phase.

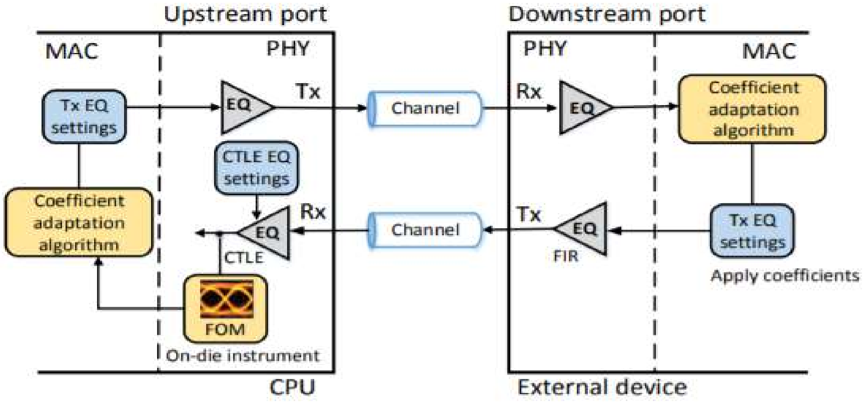


Fig. 3-6 PCIe Tx/Rx adaptative equalization. Figure taken from [17].

- Phase 2: Upstream port adjusts the Tx equalization settings of the downstream port together with its Rx settings independently for each of the lanes on the link. This phase must ensure that the Rx at upstream port have a BER < 10^{-6}.
- Phase 3: Similar to Phase 2, but the downstream port adjusts the Tx equalization settings of the upstream port together with its Rx settings independently for each of the lanes on the link. This phase must ensure that the Rx at downstream port has a BER < 10^{-6}.

3.3. Conclusions

The need for equalization as become critical due to the evolution of the communication technologies. New high speed SerDes developments have reached higher data rates above 64 Gb/s, forcing to adopt PAM4 as the signaling scheme, leading to new challenges in the channel interconnections. In order to ensure an optimal PCIe link performance, we need to find out the best EQ knobs values for Tx (FFE) and Rx (CTLE, DFE) being this a time a consuming task. Therefore, development of new methodologies to find out the best EQ settings is required. In the next chapter, we propose a new methodology aimed to tune the Tx and Rx equalizers of a PAM4 SerDes.

4. PAM4 PHY Optimization

PCIe Gen6 will adopt PAM4 modulation scheme. Above 25 Gbps, eye diagram and BER performances as FOM may not be applied due to the intrinsic limitation of receiving a closed eye diagram at the receiver, and then channel operating margin (COM) is used has the new figure of merit. We are looking to find out the optimal Tx 4 tap FFE and Rx CTLE equalizers coefficients applying a numerical optimization method [18],[19]. The procedure implies defining an objective function to maximize the COM and considering some EQ maps constraints based on the PCIe6 specification.

4.1. Channel operating margin (COM)

Receiver sensitivity cannot be definitively determined with frequency domain characteristics. Another hurdle for frequency domain analysis is that a DFE receiver is neither linear nor time invariant [20].

IEEE specification has solved the unification of transmitter, receiver, and channel specification by abandoning frequency domain based informative channel parameters for time domain specification of the system. The time domain-based specification is known as COM. The COM computation algorithm is a statistical simulation of victim and aggressor unit interval pulse responses obtained from channel scattering parameters. Reference transmitter and receiver models that include package representation and parasitic die loads are also considered in the simulation. Channel margin is a die-to-die figure of merit, as shown in Fig. 4-1, and provided as a ratio of sampled available signal to noise with a certain threshold established to mitigate discrepancy between reference chip design and actual chip design performance [20].

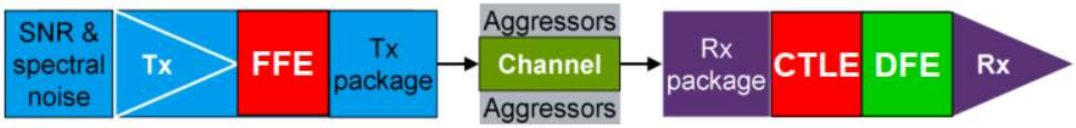


Fig. 4-1 Channel operating margin characterization.

TABLE 4-1

TX PRESET RATIOS AND CORRESPONDING COEFFICIENT VALUES FOR 64GT/S [18]

Preset #	Preshoot 2 (dB)	Preshoot 1 (dB)	De-emphasis (dB)	*c-2*	*c-1*	*c+1*	Va/Vd	Vb/Vd	Vc1/Vd	Vc2/Vd
Q0	0.0 ±0.5 dB	0.0 ±0.5 dB	0.0 ±0.5 dB	0.000	0.000	0.000	1.000	1.000	1.000	1.000
Q1	0.0 ±0.5 dB	1.6 ±0.5 dB	0.0 ±0.5 dB	0.000	-0.083	0.000	0.834	0.834	1.000	0.834
Q2	0.0 ±0.5 dB	3.5 ±0.5 dB	0.0 ±0.5 dB	0.000	-0.167	0.000	0.666	0.666	1.000	0.666
Q3	0.0 ±0.5 dB	0.0 ±0.5 dB	-1.6 ±0.5 dB	0.000	0.000	-0.083	1.000	0.834	0.834	0.834
Q4	0.0 ±0.5 dB	3.5 ±0.5 dB	-3.5 ±0.5 dB	0.000	-0.125	-0.125	0.750	0.500	0.750	0.500
Q5	0.0 ±0.5 dB	0.0 ±0.5 dB	-3.5 ±1.0 dB	0.000	0.000	-0.167	1.000	0.666	0.666	0.666
Q6	0.0 ±0.5 dB	2.9 ±0.5 dB	-6.0 ±1.0 dB	0.000	-0.083	-0.208	0.834	0.418	0.584	0.418
Q7	-1.4 ±0.5 dB	4.7 ±1.0 dB	0.0 ±0.5 dB	0.042	-0.208	0.000	0.584	0.584	1.000	0.500
Q8	-1.6 ±0.5 dB	3.5 ±0.5 dB	-3.5 ±0.5 dB	0.042	-0.125	-0.125	0.750	0.400	0.750	0.416
Q9	-1.4 ±0.5 dB	0.0 ±0.5 dB	-4.7 ±1.0 dB	0.042	0.000	-0.208	1.000	0.584	0.584	0.500
Q10	0.0 ±0.5 dB	0.0 ±0.5 dB	Note 2	0.000	0.000	Note 2	1.000	Note 2	Note 2	Note 2

Notes:

1. Reduced swing signaling must implement presets Q0, Q1, Q2, Q3, and Q5. Full swing signaling must implement all the above presets.
2. Q10 boost limits are not fixed, since its de-emphasis level is a function of the LF level that the Tx advertises during training. Q10 is used for testing the boost limit of Transmitter at full swing. Q5 is used for testing the boost limit of Transmitter at reduced swing.

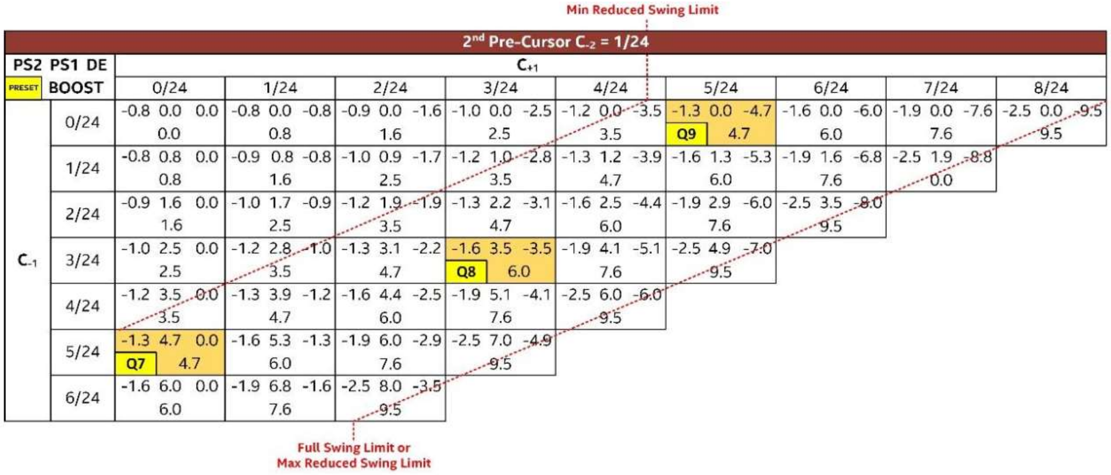
4.2. Objective function formulation and optimization

PCIe6 Tx equalization coefficients for 64 Gb/s are based on the following FIR filter relationship as shown in Fig. 3-1. The equalization coefficients are subject to constraints limiting their max swing to ±unity with C_{m2} being zero or positive, C_{m1} and C_{p} being zero or negative.

$$\left|C_{\mathrm{m2}}\right|+\left|C_{\mathrm{m1}}\right|+\left|C_0\right|+\left|C_{\mathrm{p}}\right|=1 \text{ subject to } C_{\mathrm{m2}}>0,\, C_{\mathrm{m1}}\le 0,\, C_{\mathrm{p}}\le 0 \qquad \text{(4-1)}$$

These constraints are implemented by determining only C_{m1} and C_{p} to fully define v_{out} as explained in Chapter 3 (Fig. 3-1), being C_0 implied by equation (4-2). Additionally, the coefficients range and tolerance are constrained by some requirements as follows:

- The coefficients must support all eleven values for the presets, and their respective tolerances as defined by the Tx preset ratios (Table 4-1) in the PCIe specification [21].
- The coefficients must support all eleven values for the presets, and their respective tolerances as defined by the Tx preset ratios (Table 4-1) in the PCIe specification [21].



Min Reduced Swing Limit

2nd Pre-Cursor C_{-2} = 1/24

PS2 PS1 DE		C_{+1}								
PRESET	BOOST	0/24	1/24	2/24	3/24	4/24	5/24	6/24	7/24	8/24
C_{-1}	0/24	-0.8 0.0 0.0 0.0	-0.8 0.0 -0.8 0.8	-0.9 0.0 -1.6 1.6	-1.0 0.0 -2.5 2.5	-1.2 0.0 -3.5 3.5	-1.3 0.0 -4.7 Q9 4.7	-1.6 0.0 -6.0 6.0	-1.9 0.0 -7.6 7.6	-2.5 0.0 -9.5 9.5
	1/24	-0.8 0.8 0.0 0.8	-0.9 0.8 -0.8 1.6	-1.0 0.9 -1.7 2.5	-1.2 1.0 -2.8 3.5	-1.3 1.2 -3.9 4.7	-1.6 1.3 -5.3 6.0	-1.9 1.6 -6.8 7.6	-2.5 1.9 -8.8 0.0	
	2/24	-0.9 1.6 0.0 1.6	-1.0 1.7 -0.9 2.5	-1.2 1.9 -1.9 3.5	-1.3 2.2 -3.1 4.7	-1.6 2.5 -4.4 6.0	-1.9 2.9 -6.0 7.6	-2.5 3.5 -8.0 9.5		
	3/24	-1.0 2.5 0.0 2.5	-1.2 2.8 -1.0 3.5	-1.3 3.1 -2.2 4.7	-1.6 3.5 -3.5 Q8 6.0	-1.9 4.1 -5.1 7.6	-2.5 4.9 -7.0 9.5			
	4/24	-1.2 3.5 0.0 3.5	-1.3 3.9 -1.2 4.7	-1.6 4.4 -2.5 6.0	-1.9 5.1 -4.1 7.6	-2.5 6.0 -6.0 9.5				
	5/24	-1.3 4.7 0.0 Q7 4.7	-1.6 5.3 -1.3 6.0	-1.9 6.0 -2.9 7.6	-2.5 7.0 -4.9 9.5					
	6/24	-1.6 6.0 0.0 6.0	-1.9 6.8 -1.6 7.6	-2.5 8.0 -3.5 9.5						

Full Swing Limit or
Max Reduced Swing Limit

Fig. 4-2 Transmit Equalization Coefficient Space Triangular Matrix Example for 64GT/s. Figure taken from [21].

- In order to keep the output transmitted power constant with respect to coefficients, the *FS* which indicates the maximum differential voltage that can be generated by the Tx is defined as:

$$FS = |C_{\mathrm{m1}}| + |C_{\mathrm{m2}}| + |C_0| + |C_{\mathrm{p}}| \tag{4-3}$$

- The flat level voltage should always be greater than the minimum differential voltage that can be generated by the Tx indicated as the *LF* parameter,

$$|C_0| - |C_{\mathrm{m1}}| - |C_{\mathrm{m2}}| - |C_{\mathrm{p}}| \geq LF \tag{4-4}$$

When all the above PCIe specification constraints are applied, the resulting coefficients space may be mapped onto a triangular matrix, as shown in Fig. 4-2 [21].

Considering the Tx and Rx equalizers as part of the PCIe adaptative equalization process as described in Section 3.2 multiple superimposed EQ maps can be built, one per CTLE coefficient (C_{ctle}) value as shown in Fig. 4-3. C_{m1} and C_{p} coefficients are mapped onto the y-axis and x-axis, respectively. Each matrix cell corresponds to a valid combination of C_{m1} and C_{p} coefficients, and $u(\boldsymbol{x}^*)$ correspond to a combination of C_{m1}, C_{p} and C_{ctle} that results in an eye diagram qualified as optimum as explained later in greater detail. This EQ maps can be used as an intuitive visual indicator of the equalization performance.

The current post-silicon method for Gen3-Gen5 to find the best subset of coefficients for both Tx and Rx, consists of using these EQ maps, which are obtained by measuring the eye

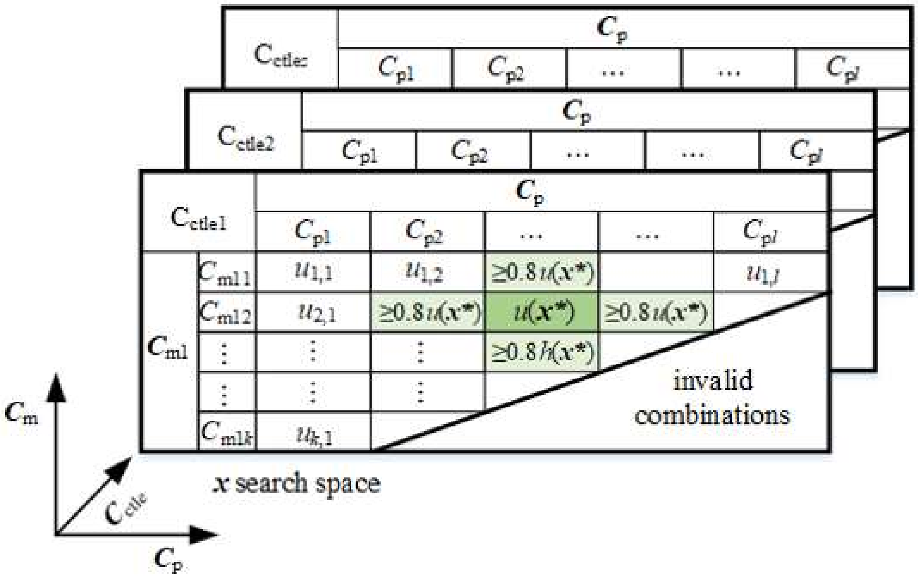


Fig. 4-3 PCIe EQ map coefficients search space for optimization. Figure taken from [9].

diagram characteristics as FOM (i.e. eye height, eye width, or eye diagram area) of the received signal for each of the C_m, C_p, and C_{ctle} combinations across each lane and device pairing, requiring multiple EQ maps. The method consists of finding the set of Tx and Rx coefficients that qualify the FOM as near optimal. However, this has to be performed by ensuring at the same time that the responses around the best Cm-Cp matrix cell are at least 80% of the value of that matrix cell, as illustrated in Fig. 4-3, to avoid selecting a combination of too-high sensitivity.

Due to the large number of EQ maps, along with all the new challenges imposed by PAM4, finding the optimal subset of coefficients would be a very challenging task.

COM is a signal to noise ratio, and it is a new figure of merit (FOM) that considers the passive and active channel components. COM has been adopted by several standards of communications, and it is gaining attention as a valuable tool for analyzing high-speed digital channels, especially for signaling speeds beyond 25 Gb/s. Above such data rate, eye diagram and BER performances as FOM may not be applied due to the intrinsic limitation of receiving a closed eye diagram at the receiver [22].

COM is a ratio between a calculated signal amplitude to a calculated noise amplitude [23] defined as,

$$COM = 20\log\frac{A_{signal}(x)}{A_{noise}} \tag{4-5}$$

where A_{signal} is a signal with a data rate of PCIe Gen6 and A_{noise} is the noise on the signal considering inter-symbol interference (ISI), random and dual-Dirac jitter noise, and crosstalk. We aim at finding the optimal set of coefficients values of the Tx and Rx equalizers (C_{m1}, C_{p}, C_{ctle}) contained in the vector $\boldsymbol{x}$ to maximize COM. Therefore, an initial objective function to be minimized is defined as,

$$u(x) = -20\log\frac{A_{signal}(x)}{A_{noise}} \tag{4-6}$$

The signal amplitude, A_{signal}, comes from the middle eye [24], and the combined noise term, A_{noise}, is the vertical eye closure as shown in the Fig. 4-4. The optimization problem for COM is then defined as,

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} u(\boldsymbol{x}) \tag{4-7}$$

with $u(\boldsymbol{x})$ defined by (4-8).

We need to ensure the optimal system response is within a suitable area in the coefficients search space of the EQ map as shown in Fig. 4-3. Here we follow our work in [2] to define the corresponding objective function. The four responses around $u(\boldsymbol{x}^*)$ must be at least 80% of the value of $u(\boldsymbol{x}^*)$, as shown in Fig. 4-3, where $u_{i,j}$ are the objective function values per (4-9) for the i-th C_{m1} and j-th C_{p} values, and

$\boldsymbol{C}_{\text{m}} = [C_{\text{m1}} \ \cdots \ C_{\text{m}k}]^{\text{T}}$ is the vector of Tx FIR pre-cursor values (4-10)

$\boldsymbol{C}_{\text{p}} = [C_{\text{p1}} \ \cdots \ C_{\text{p}l}]^{\text{T}}$ is the vector of Tx FIR post-cursor values (4-11)

$\mathbf{C}_{\text{ctle}} = [C_{\text{r1}} \ \cdots \ C_{\text{r}z}]^{\text{T}}$ is the vector of Rx CTLE coefficient values (4-12)

The new optimization problem can be defined through a constrained formulation, such that the optimal set of coefficients maximizes the system response without exceeding the limit of $0.8u(x^*)$ in the vicinity,

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} u(\boldsymbol{x}) \ \text{subjectto} \ l_1(\boldsymbol{x}) \le 0, l_2(\boldsymbol{x}) \le 0, l_3(\boldsymbol{x}) \le 0, l_4(\boldsymbol{x}) \le 0 \tag{4-13}$$

With

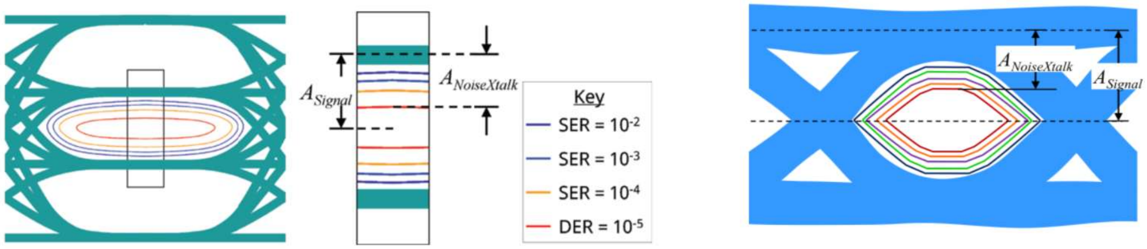


Fig. 4-4 An eye diagram calculated from the pulse response of a transmitter-channel-receiver model with COM variables. Figure taken from [24].

$$l_1(x) = 0.8u(C_{\mathrm{ml}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{ml}i*+1}, C_{\mathrm{p}j*}) \tag{4-14}$$

$$l_2(x) = 0.8u(C_{\mathrm{ml}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{ml}i*-1}, C_{\mathrm{p}j*}) \tag{4-15}$$

$$l_3(x) = 0.8u(C_{\mathrm{ml}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{ml}i*}, C_{\mathrm{p}j*+1}) \tag{4-16}$$

$$l_4(x) = 0.8u(C_{\mathrm{ml}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{ml}i*}, C_{\mathrm{p}j*-1}) \tag{4-17}$$

where $C_{\mathrm{m}1i*}$ and $C_{\mathrm{p}i*}$ are the set of coefficients that maximize the FOM for each of the C_{ctle} values. A more convenient unconstrained formulation can be defined by adding a penalty term,

$$U(x) = -20\log\frac{A_{signal}(x)}{A_{noise}} + |L(x)|^2\left[\frac{|u(x^{(0)})|}{\left|\max\left\{l(x^{(0)})\right\}\right|^2}\right] \tag{4-18}$$

where $L(\boldsymbol{x})$ is a corner limits penalty function, defined as,

$$L(\boldsymbol{x}) = \max\{0, l_1(\boldsymbol{x}), l_2(\boldsymbol{x}), l_3(\boldsymbol{x}), l_4(\boldsymbol{x})\} \tag{4-19}$$

and $x^{(0)}$ is the starting point. Then, our unconstrained objective function to optimize the system response is

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} U(\boldsymbol{x}) \tag{4-20}$$

With $U(\boldsymbol{x})$ defined by (4-21).

The optimal set of coefficients values $\boldsymbol{x}^*$ is found by solving (4-22) and using a low-cost computational optimization technique such as the Nelder-Mead method without the need of estimating gradients.

4.3. Simulation environment

To implement the proposed methodology, we use the SerDes Toolbox™ [25] to simulate a PCIe6 SerDes. This Toolbox provides a Matlab and Simulink model library and a set of analysis tools and apps for the design and verification of SerDes systems. By exploiting this Matlab model, the need of developing a metamodel based on actual measurements is avoided. [26], [27].

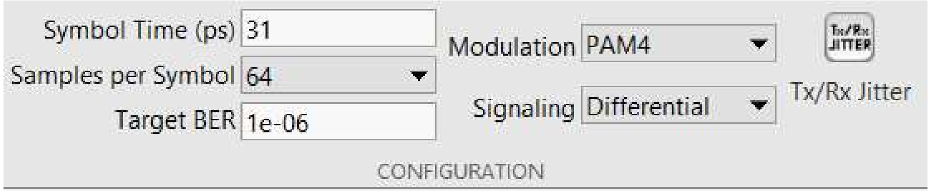
With the SerDes Designer app, statistical analysis can be used to rapidly design wired communications links. The app provides parameterized models and algorithms that let to explore a wide range of equalizer configurations to improve channel performance. Metrics such as eye diagram, bathtub curve, and COM, including the effects of jitter and crosstalk can be assessed.

With Matlab based building blocks such as CTLE, DFE, FFE, and CDR, a chosen architecture can be described using datasheets or measurement data and simulate control and adaptive algorithms.

SerDes toolbox has a predefined model with analog output, channel and analog input blocks. In order to build the PCIe6 model the settings described in Fig. 4-5 are required. The symbol time should be set to 31ps to simulate a PCIe Gen6 SerDes running with a data transfer of 64GT/s, the modulation has to be set at PAM4, the signaling into differential and the BER according to PCIe specification need to be set at $10e^{-6}$.

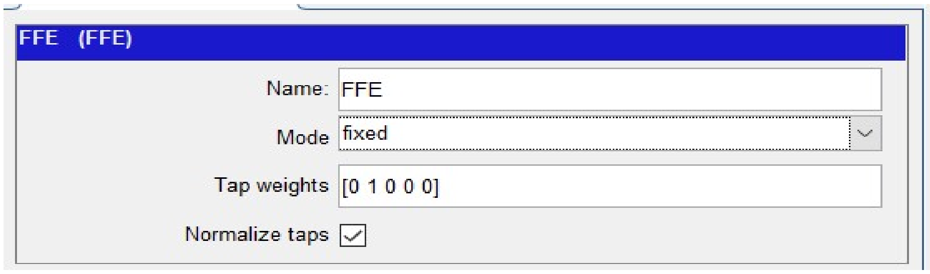
The PCIe Gen 6 specification defines a Feed Forward Equalizer for the transmitter which consists of a four-tap weight equalizer. The toolbox has the capability to manipulate the number of taps and the weights, and it has two modes (fixed and off). This option provides the capability to manipulate the behavior of the FEE as shown in Fig. 4-6.

The toolbox has multiple options to manipulate the channel configuration. The first one is the channel loss, for our simulation we use 36db channel to cover the maximum channel loss of a PCIe6 interconnect. The next option is the differential impedance in Ohms, for our simulation we



Symbol Time (ps) 31
Samples per Symbol 64
Target BER 1e-06
Modulation PAM4
Signaling Differential

Fig. 4-5 SerDes system configuration.



FFE (FFE)

Name: FFE

Mode fixed

Tap weights [0 1 0 0 0]

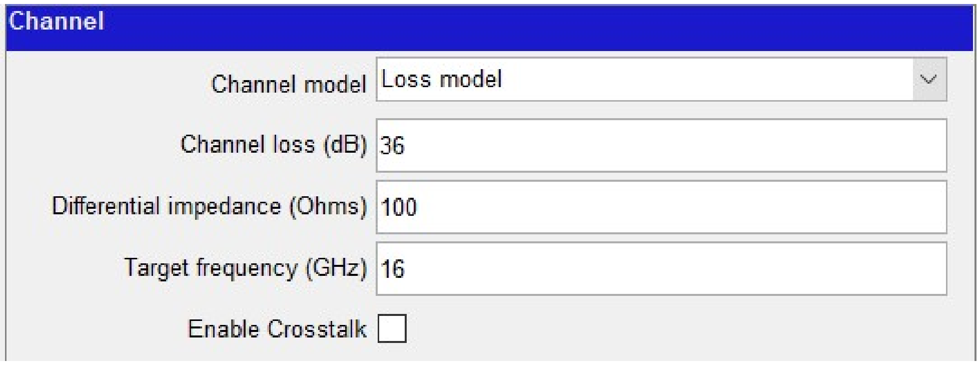
Normalize taps ☑

Fig. 4-6 Transmitter FFE configuration.

use 100 Ohms which is defined in the PCIe Gen6 specification, and the last option is the target frequency which is defined in Ghz to cover the target frequency for PCIe6 we use 16Ghz as shown in Fig. 4-7.

The receiver equalization architecture is modeled in the toolbox. It has the capability to implement multiple equalizers such as the variable gain amplifier (VGA), automatic gain control (AGC), CTLE, DFE, and the CDR. In our simulation we define three equalization stages, the first one is the VGA which is set to automatic in the toolbox, and this means the simulation will take the best value to equalize the signal. The second stage is the CTLE, and the toolbox has the capability to change the DC gain and peaking gain of it. The toolbox has also the capability to change the peaking frequency, and then in order to meet the PCIe Gen6 data rate a frequency of 16Ghz is set for the peaking.

We define 25 taps for the DC gain and peaking, then the CTLE equalizes from 0dB to -25dB as show in Fig 3-4.



Channel

Channel model Loss model

Channel loss (dB) 36

Differential impedance (Ohms) 100

Target frequency (GHz) 16

Enable Crosstalk ☐

Fig. 4-7 Channel configuration.

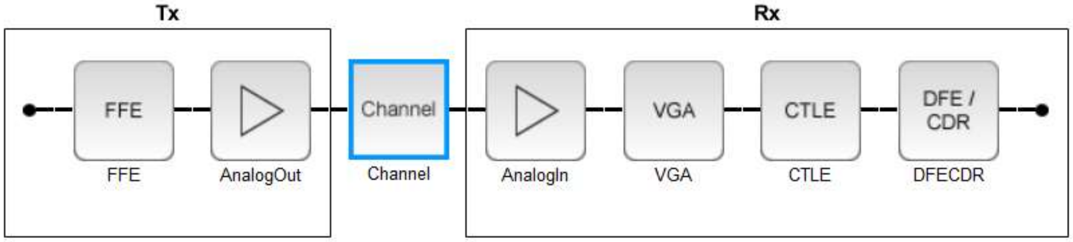


Fig. 4-8 SerDes equalization block diagram.

The last block defined in the receiver architecture is the DFE/CDR (see Fig. 4-8.) which is set of 18 tap weights based on recently design literature [28].

To generate the Matlab code for each block of the SerDes, the toolbox has an option to export the code to Simulink or Matlab, for our purpose we will choose the second option as shown in Fig. 4-9. Once the SerDes design is in Matlab code this is divided in functions for easy manipulation during the optimization process as shown in the flow diagram (see Fig. 4-10). The complete code is listed in the appendix A.

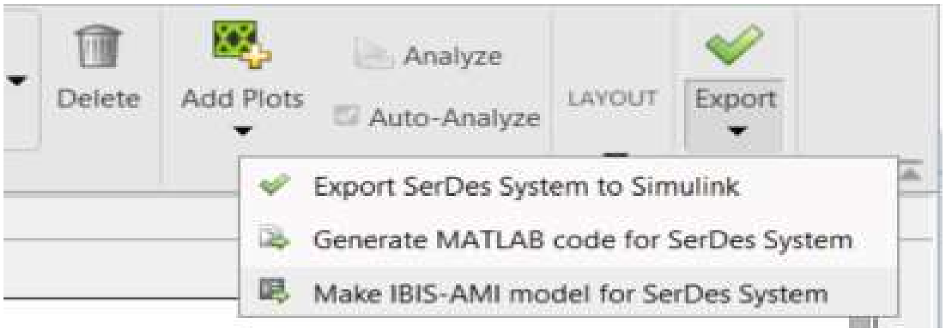

Delete
Add Plots
Analyze
Auto-Analyze
LAYOUT
Export
Export SerDes System to Simulink
Generate MATLAB code for SerDes System
Make IBIS-AMI model for SerDes System

Fig. 4-9 Export SerDes System to MATLAB Code.

4. PAM4 PHY OPTIMIZATION

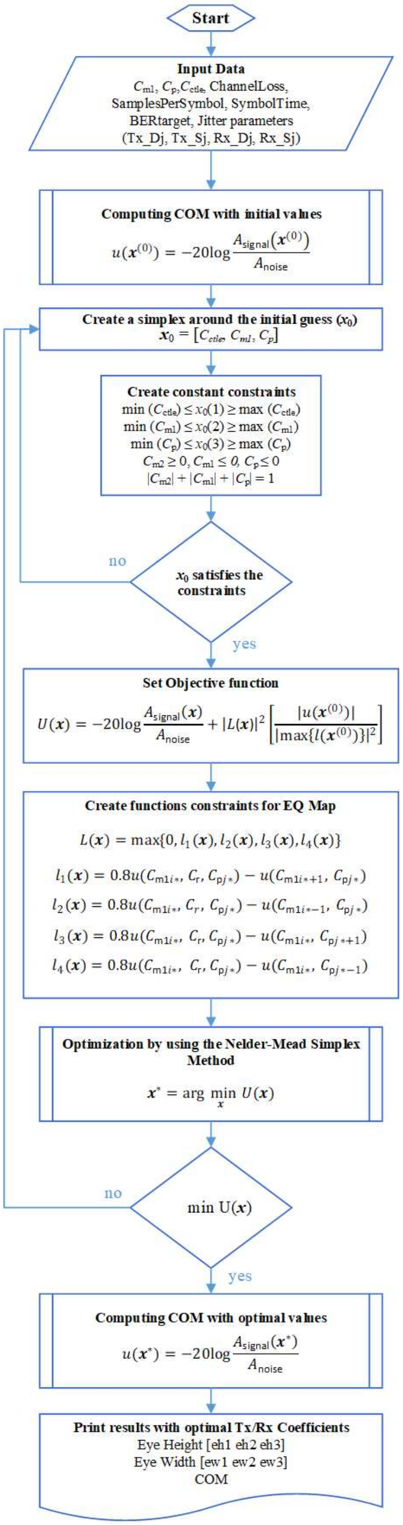

Start
Input Data
C_{m1}, C_{p}, C_{ctle}, ChannelLoss,
SamplesPerSymbol, SymbolTime,
BERtarget, Jitter parameters
(Tx_Dj, Tx_Sj, Rx_Dj, Rx_Sj)
Computing COM with initial values
$u(\boldsymbol{x}^{(0)}) = -20\log\frac{A_{\mathrm{signal}}(\boldsymbol{x}^{(0)})}{A_{\mathrm{noise}}}$
Create a simplex around the initial guess (x_0)
$\boldsymbol{x}_0 = [C_{ctle}, C_{m1}, C_p]$
Create constant constraints
$\min(C_{\mathrm{ctle}}) \leq x_0(1) \geq \max(C_{\mathrm{ctle}})$
$\min(C_{\mathrm{m1}}) \leq x_0(2) \geq \max(C_{\mathrm{m1}})$
$\min(C_{\mathrm{p}}) \leq x_0(3) \geq \max(C_{\mathrm{p}})$
$C_{\mathrm{m2}} \geq 0, C_{\mathrm{m1}} \leq 0, C_{\mathrm{p}} \leq 0$
$|C_{\mathrm{m2}}| + |C_{\mathrm{m1}}| + |C_{\mathrm{p}}| = 1$
no
x_0 satisfies the constraints
yes
Set Objective function
$U(\boldsymbol{x}) = -20\log\frac{A_{\mathrm{signal}}(\boldsymbol{x})}{A_{\mathrm{noise}}} + |L(\boldsymbol{x})|^2 \left[\frac{|u(\boldsymbol{x}^{(0)})|}{|\max\{l(\boldsymbol{x}^{(0)})\}|^2}\right]$
Create functions constraints for EQ Map
$L(\boldsymbol{x}) = \max\{0, l_1(\boldsymbol{x}), l_2(\boldsymbol{x}), l_3(\boldsymbol{x}), l_4(\boldsymbol{x})\}$
$l_1(\boldsymbol{x}) = 0.8u(C_{\mathrm{m1}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{m1}i*+1}, C_{\mathrm{p}j*})$
$l_2(\boldsymbol{x}) = 0.8u(C_{\mathrm{m1}i*}, C_r, C_{\mathrm{p}j*}) - u(C_{\mathrm{m1}i*-1}, C_{\mathrm{p}j*})$
$l_3(\boldsymbol{x}) = 0.8u(C_{\mathrm{m1}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{m1}i*}, C_{\mathrm{p}j*+1})$
$l_4(\boldsymbol{x}) = 0.8u(C_{\mathrm{m1}i*}, C_{\mathrm{r}}, C_{\mathrm{p}j*}) - u(C_{\mathrm{m1}i*}, C_{\mathrm{p}j*-1})$
Optimization by using the Nelder-Mead Simplex Method
$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} U(\boldsymbol{x})$
no
min U($\boldsymbol{x}$)
yes
Computing COM with optimal values
$u(\boldsymbol{x}^*) = -20\log\frac{A_{\mathrm{signal}}(\boldsymbol{x}^*)}{A_{\mathrm{noise}}}$
Print results with optimal Tx/Rx Coefficients
Eye Height [eh1 eh2 eh3]
Eye Width [ew1 ew2 ew3]
COM

Fig. 4-10 . SerDes Optimization Flow Diagram.

4.4. Results

Following the optimization process defined in Section 4.2, we found a set of Tx and Rx coefficients that minimize the objective function *U(**x**)* in just 160 evaluations as shown in Fig. 4-11. The time consumed by the optimization process was 33.4 minutes.

Fig. 4-12 shows the evolution of the Tx (C_{m1}, C_{p}) and Rx (C_{ctle}) coefficients during the optimization process.

Fig. 4-13 shows the eye diagram before the optimization process, showing asymmetry issues and low values of eye height and eye width in the three eye diagrams. Fig. 4-14 shows the eye diagrams after the optimization at the receiver using the EQ coefficients from the optimization process. The optimized equalization coefficients yield an eye width and eye height improvements of 27% average and 131% in average respectively, and a COM improvement of 150%, reducing also the eyes asymmetries. The obtained eye-diagram results confirm the effectiveness of the proposed approach.

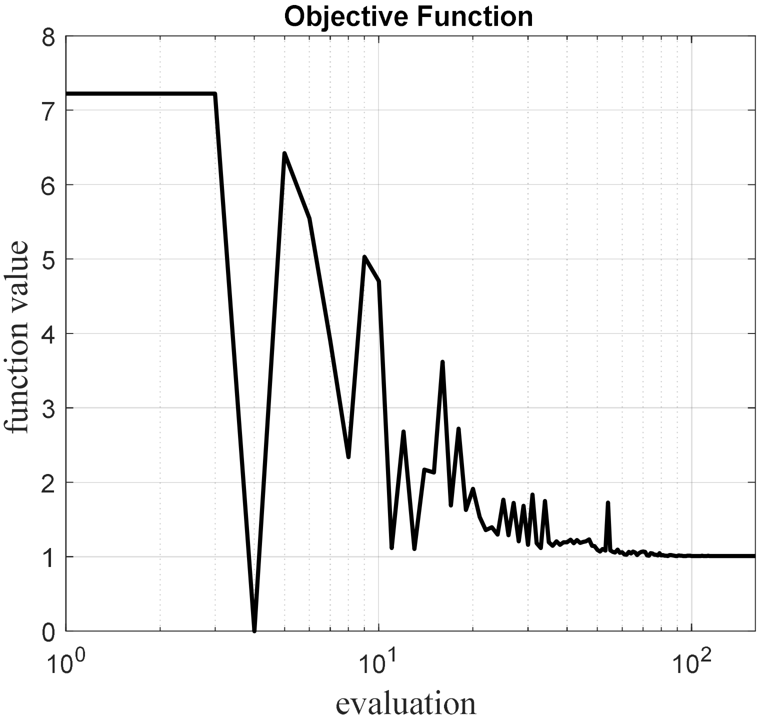


Fig. 4-11 Optimization process evolution.

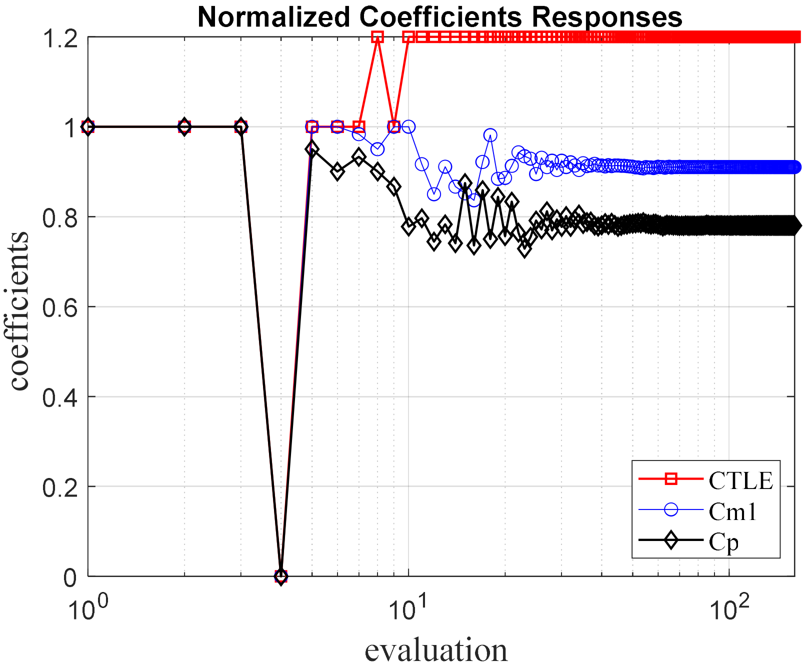


Fig. 4-12 Tx and Rx coefficients evolution over the optimization process.

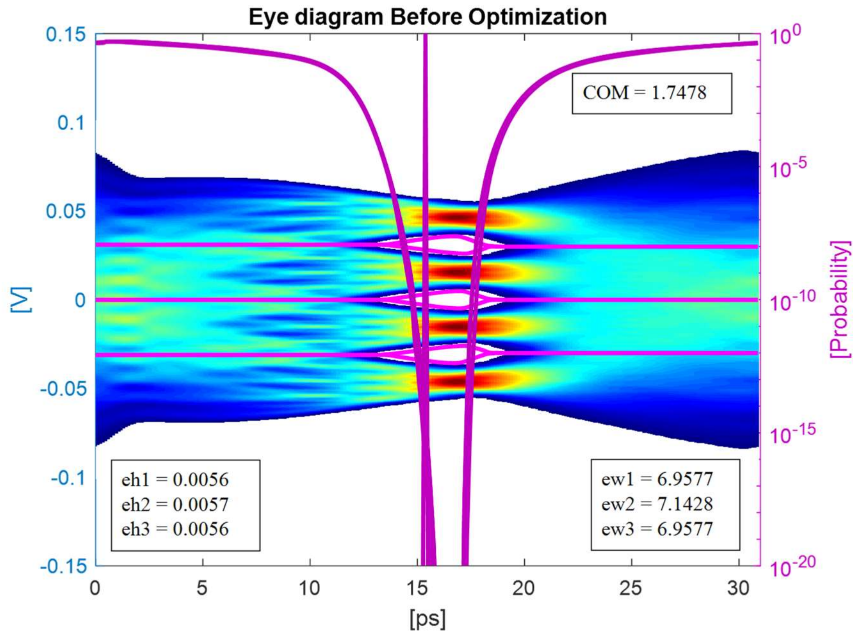


Fig. 4-13 Eye diagram before the optimization process.

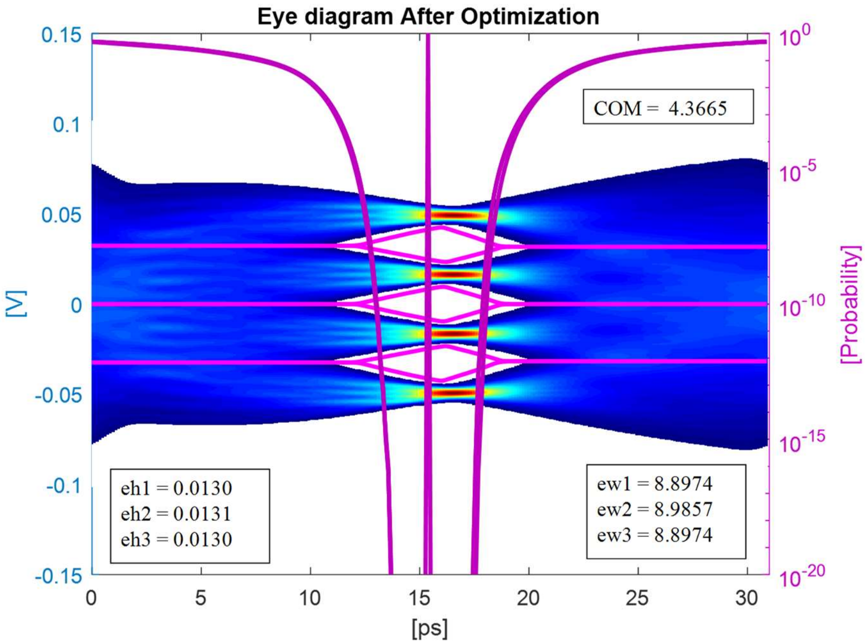


Fig. 4-14 Eye diagram results after the optimization process.

5. General Conclusions

The demand of higher data rates is leading to develop new PCIe generations; this is bringing new challenges in post-silicon validation. The complexity and performance of the new equalizers in PCIe products is demanding a lot of time during the equalization process in post-silicon validation. In order to reduce the time, we propose a methodology to optimize the Tx and Rx equalizers in PCIe6 and improve the link performance.

In Chapter 1, a brief overview in PCIe evolution was presented. The need to increase the PCIe bandwidth through generations was discussed, leading to the adoption of PAM4 to satisfy the demand of multiple computer markets. The pulse amplitude modulation 4 levels scheme was discussed on details in Chapter 2.

In Chapter 3, an overview of the PCIe Equalization architecture was presented, concluding that the equalization plays an important role even with PAM4 signaling. Tx and Rx equalization schemes such de-emphasis/pre-emphasis for Tx and CTLE/DFE are widely used in high-speed serial links and they continue to be used for PAM4.

An efficient methodology to optimize the Tx 4 tap FFE and the Rx CTLE equalizers applying a direct numerical optimization based on Nelder-Mead was proposed in Chapter 4. The procedure implies defining an objective function to maximize the channel operating margin (COM) and considering some EQ maps constrains based on the PCIe6 specification.

While there are not still silicon samples with PCIe6, the evaluation of the proposed method was performed by using the MATLAB SerDes Toolbox. The subset of coefficients found during the optimization showed a substantial improvement in COM and eye-diagram.

The methodology presented in this thesis will allow to accelerate the process of finding the best subset of EQ knobs that improves the PCIe6 link. The methodology can also be implemented in other SerDes interfaces with PAM4 such as ethernet.

This thesis dissertation offers the possibility of a number of future research opportunities. One possible future work consists of using the proposed optimization techniques to optimize even more the PCIe6 SerDes system by looking to improve the CTLE and DFE equalizers circuits. The number of stages and taps on these two equalizers can be optimized to improve the response of the system. The temperature effect can be also added to the SerDes model to optimize the equalizers looking to mitigate the effects caused by the environment temperature, especially for PCIe applications where the product is exposed to extreme temperature settings such as industrial and automotive applications.

Appendix

A. MATLAB CODE

PAM4 main script

```
% PAM4EQOptimization.m
%==================================================
% PCIeGen6 - PAM4Equalizers Optimization
% Francisco E. Rangel-Patino and Roberto J. Ruiz-Urbina
% Research Group on Computer-Aided Engineering of Circuits and Systems
% (CAECAS)
% Department of Electronics, Systems, and Informatics ITESO
% Intel Guadalajara Design Center - Electrical Validation
% January 2021
% Version V12
%==================================================
% management functions
clear  % clear all variable/information in the workspace
clear global  %  again use caution - clears global information
clc    % position the cursor at the top of the screen
format compact  % avoid skipping a line when writing to the command window
warning off  % don't report any warnings like divide by zero etc
%==================================================
%                  DESIGN VARIABLES
%%% Define CTLE and FIR Filter Coefficients Vectors per PCIeGen6 Spec
% Cm2 = 1/24
% x1: CTLE = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
% x2: Cm1 = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24]
% x3: Cp  = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24 7/24 8/24]
%     C0  = 1 - |Cm1| - |Cp| - |Cm2|
% Define TapWeights Vector for SerDes ToolBox Simulator
% txBlocks{1}.TapWeights = [0 Cm2 Cm1 C0 Cp]
%==================================================
%                  SERDES SYSTEM SETUP
%<We will use integers numbers for Cm, Cp and CTLE values for optimization, but will convert to
actual values (x/24) for simulation>
Cm2   = 1/24; % Tx FIR Filter coefficient Cm2
Cm1V  = -1*[0 1 2 3 4 5 6]/24; % Tx FIR Filter coefficient Cm1
CpV   = -1*[0 1 2 3 4 5 6 7 8]/24; % Tx FIR Filter coefficient Cp
CTLEV = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]; % Rx CTLE values
% ChannelLoss: Channel losses value
ChannelLoss = 36;%25;
% SamplesPerSymbol: Number of data points per symbol.
% The Samples per symbol determine the acquisition bandwidth. PCIeGen6 Data Rate (Gb/s) = 64.0
SamplesPerSymbol = 64;
%  ModulationLevels: Number of logic levels in the modulation scheme: PAM4
ModulationLevels = 4;
% SymbolTime: Time it takes to send one symbol across the link.
% Per PCIeBase Specification for  PCIe Gen6: Unit Interval
% (UI(Tx))=31.246875 psec
SymbolTime = 31e-12;
% BERtarge: Target bit error rate. Target bit error rate used to generate eye-contours,
% specified as a unitless real positive scalar. PAM4 requires higher BER at the physical layer
(~1e-6)
BERtarget = 1e-06;
% Jitter Configuration: Jitter parameters defined as Type UI:
Tx_Dj = .5e-12; % Tx Deterministic jitter
Tx_Sj = .3e-12; % Tx sinusoidal jitter
Rx_Dj = .4e-12; % Rx Deterministic jitter
Rx_Sj = .7e-12; % Rx sinusoidal jitter
%
%==================================================
%                  Recording starting time
    ti = clock;
    clc
    display(['Starting optimization at [HH:MM:SS]:' datestr(now,'HH:MM:SS')])
    fprintf('Running... \n')
```

```
%==================================================
%                   DIRECT OPTIMIZATION
% Nelder-Mead Algorithm options configuration
MaxIter = 1e4;
MaxFunEvals = 1e6;
tolx=1e-6;
tolfun=1e-6;
options = optimset('MaxFunEvals',MaxFunEvals,'MaxIter',MaxIter,'TolX',tolx,'TolFun',tolfun);
%---------------------------------------------------

% We will use the fminsearch matlab function for Nelder-Mead optimization.
% fminsearch  call “PAM4_Fun_Opt” to extract the eye diagram measurements (simulations) and find
out the Cm, Cp, and CTLE values
% that minimize the objective function
 Xo = [15,-0.25,-0.083333]; % Seed: Xo = [CTLE, Cm1, Cp]

 % Computing objective functin "u0(Xo)" (with initial values)
 Ctle = Xo(1);
 Cm1 = Xo(2);
 Cp = Xo(3);
 C0 = 1 - abs(Cm1) - abs(Cp) - abs(Cm2);
 [u,EHj,EWj,PAM4_systems_Jitter] =
PAM4_SerDes(Cm2,Cm1,Cp,Ctle,C0,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BERtarget
,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
 u0 = u;
 Com0 = abs(u);
 EHj0 = EHj;
 EWj0 = EWj;

 %=======================================
 % ploting Eye Diagram Before optimization

 figure(1)
 plotStatEye(PAM4_systems_Jitter)
 title('Eye diagram Before Optimization');

%==================================================
%   Global variables definition to register numer of models evaluations
    global fevaluations U_i ctle_i Cm1_i Cp_i
    fevaluations = 0;
%==================================================
 % Direct Optimization Procedure using Nelder-Mead Simplex Method (fminsearch)

 [Xopt,fval] =
fminsearch(@PAM4_Fun_Opt,Xo,options,u0,Cm2,Cm1V,CpV,CTLEV,ChannelLoss,SamplesPerSymbol,Modulation
Levels,SymbolTime,BERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
              [knob] = knobfilter(CTLEV',Xopt(1)); % Rounding Xo(1) to a valid value per CTLEV
vector
                      Xopt(1) = knob;
              [knob] = knobfilter(Cm1V',Xopt(2)); % Rounding Xo(1) to a valid value per Cm1
vector
                      Xopt(2) = knob;
              [knob] = knobfilter(CpV',Xopt(3)); % Rounding Xo(1) to a valid value per Cp vector
                      Xopt(3) = knob;

CTLEb = Xopt(1); % Optimal Rx CTLE coefficient values
Cm1b  = Xopt(2); % Optimal Tx Cm1 coefficient values
Cpb   = Xopt(3); % Optimal Tx Cp coefficient values
fsol  = fval;
%==================================================

 C0opt = 1 - abs(Cm1b) - abs(Cpb) - abs(Cm2);
 [u,EHj,EWj,PAM4_systems_Jitter] =
PAM4_SerDes(Cm2,Cm1b,Cpb,CTLEb,C0opt,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BER
target,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
 Com_opt = abs(u);
 EHj_opt = EHj;
 EWj_opt = EWj;
 %=======================================
 % ploting Eye Diagram After optimization
 figure(2)
 plotStatEye(PAM4_systems_Jitter)
 title('Eye diagram After Optimization');

%==================================================
%                   Recording finishing time
    tf = clock;
    optimizationTime = etime(tf,ti)/60;
%==================================================
%                   Printing results
clc
fprintf('Optimization finished; time elapsed: %4.1f min\n',optimizationTime)
%==================================================
fprintf('-------------------------------------------------\n')
fprintf('PCIeGen6 - PAM4 Equalizers Optimization \n')
fprintf('-------------------------------------------------\n')
fprintf('\nChannel Loss [dB] ='),disp(ChannelLoss)
fprintf('\nTx Deterministic jitter ='),disp(Tx_Dj)
fprintf('\nTx sinusoidal jitter ='),disp(Tx_Sj)
fprintf('\nRx Deterministic jitter ='),disp(Rx_Dj)
fprintf('\nRx sinusoidal jitter ='),disp(Rx_Sj)
fprintf('\nEye Height values before optimization [eh1 eh2 eh3] ='),disp(double([EHj0']))
fprintf('\nEye Width values before optimization [ew1 ew2 ew3] ='),disp(double([EWj0']))
fprintf('\nCOM value before optimization: COM ='),disp(Com0)
fprintf('\nSolution: \n')
fprintf('\nObjective function evaluations:'),disp(fevaluations)
fprintf('\nBest objective function : U ='),disp(fsol)
fprintf('\nOptimized Tx/Rx coefficients [CTLE Cm1 Cp] ='),disp(double([CTLEb Cm1b Cpb]))
fprintf('\nEye Height values with optimal Tx/Rx Coefficients [eh1 eh2 eh3]
='),disp(double([EHj_opt']))
fprintf('\nEye Width values with optimal Tx/Rx Coefficients [ew1 ew2 ew3]
='),disp(double([EWj_opt']))
fprintf('\nCOM value with optimal Tx/Rx Coefficients: COM ='),disp(Com_opt)
fprintf('\n\n')
%==================================================
%            PREPARING DATA FOR PLOT
        Iteration = 1:fevaluations;
        Ctle = ctle_i;
        Cm1 = Cm1_i;
        Cp = Cp_i;
        Function = U_i;
        Ctle_norm = Ctle/Ctle(1);
        Cm1_norm = Cm1/Cm1(1);
        Cp_norm = Cp/Cp(1);
%==================================================
%          PLOTING RESULTS
    figure
    p1 = semilogx(Iteration,Ctle_norm,'r-s'); hold on
    P2 = semilogx(Iteration,Cm1_norm,'b-o'); hold on
    p3 = semilogx(Iteration,Cp_norm,'k-d');
    p1(1).LineWidth = 1;
    p2(1).LineWidth = 1;
    p3(1).LineWidth = 1;
    hold off
    set(gca,'fontsize',12)
    xlabel('evaluation ','FontName','Times','FontSize',16);
    ylabel('coefficients ','FontName','Times','FontSize',16);
    legend('\it\rmCTLE','\it\rmCm1','\it\rmCp','location','best');
    set(legend,'FontSize',12,'FontName','Times');
    title('Normalized Coefficients Responses');
    grid on;
    %
    figure
    p1 = semilogx(Iteration,Function,'k');
    p1(1).LineWidth = 2;
    set(gca,'fontsize',12)
    xlabel('evaluation ','FontName','Times','FontSize',16);
    ylabel('function value ','FontName','Times','FontSize',16);
    %legend('\it\bfC\rm_m','\it\bfC\rm_0','\it\bfC\rm_p','location','best');
    %set(legend,'FontSize',12,'FontName','Times');
    title('Objective Function');
    grid on;
%
%==================================================
%                   COM Equalization Map
%
fprintf('Running EQMap (sweeping Cm1 and Cp) at optimal CTLE value... \n')
for C = 1:length(CpV)
        for R = 1:length(Cm1V)
           C0 = 1 - abs(Cm1V(R)) - abs(CpV(C)) - abs(Cm2);
           COM_EQmap(R,C)=
PAM4_SerDes(Cm2,Cm1V(R),CpV(C),CTLEb,C0,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,
BERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
        end
end
fprintf('Plotting EQMap... \n')
figure
contourf(CpV,Cm1V,COM_EQmap,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');  % add axis labels and plot title
ylabel('Cm1');
title('COM Equalization Map');

figure
surf(CpV,Cm1V,COM_EQmap)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp'); % add axis labels and plot title
ylabel('Cm1');
zlabel('CTLE');
title('COM Equalization Map');
```

PAM4 SerDes function

```
% PAM4_SerDes.m
%==================================================
% PCIeGen6 - PAM4 Equalizers Optimization
% Francisco E. Rangel-Patino and Roberto J. Ruiz-Urbina
% Research Group on Computer-Aided Engineering of Circuits and Systems
% (CAECAS)
% Department of Electronics, Systems, and Informatics ITESO
% Intel Guadalajara Design Center - Electrical Validation
% January 2021
% Version V11
%==================================================
%                 DESIGN VARIABLES
%%% Define CTLE and FIR Filter Coefficients Vectors per PCIeGen6 Spec
% Cm2 = 1/24
% x1: CTLE = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
% x2: Cm1 = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24]
% x3: Cp  = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24 7/24 8/24]
%     C0  = 1 - |Cm1| - |Cp| - |Cm2|
% Define TapWeights Vector for SerDes ToolBox Simulator
% txBlocks{1}.TapWeights = [0 Cm2 Cm1 C0 Cp]
%==================================================
function  [u,EHj,EWj,PAM4_systems_Jitter] =
PAM4_SerDes(Cm2,Cm1,Cp,Ctle,C0,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BERtarget
,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj)
%
% Ctle
% Cm1
% Cp
% C0
% ------------------------------------
% MATLAB script to build SerDes System
% ------------------------------------
% Build cell array of Tx blocks:
%TapWeights = [0 1 0 0 0];
txBlocks{1} = serdes.FFE;
txBlocks{1}.BlockName = 'FFE';
txBlocks{1}.Mode = 1;
% txBlocks{1}.TapWeights = [0 0.04 0.125 0.833 0];%[0 Cm2 Cm1 C0 Cp]
txBlocks{1}.TapWeights = [Cm2 Cm1 C0 Cp];%[0 Cm2 Cm1 C0 Cp]
txBlocks{1}.Normalize = true;
% ------------------------------------
% Build cell array of Rx blocks:
rxBlocks{1} = serdes.VGA;
rxBlocks{1}.BlockName = 'VGA';
rxBlocks{1}.Mode = 1;
rxBlocks{1}.Gain = 1;
rxBlocks{2} = serdes.CTLE;
rxBlocks{2}.BlockName = 'CTLE';
rxBlocks{2}.Mode = 1;
rxBlocks{2}.ConfigSelect = Ctle;% Here we will use Ctle variable to sweep CTLE vector
rxBlocks{2}.Specification = 'DC Gain and Peaking Gain';
rxBlocks{2}.PeakingFrequency = 32000000000;
rxBlocks{2}.DCGain = [0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -
21 -22 -23 -24 -25];
rxBlocks{2}.PeakingGain = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25];
rxBlocks{2}.ConfigSelect = Ctle;% Here we will use Ctle variable to sweep CTLE vector
rxBlocks{2}.Specification = 'DC Gain and Peaking Gain';
rxBlocks{2}.PeakingFrequency = 32000000000;
rxBlocks{2}.DCGain = [0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -
21 -22 -23 -24 -25];
rxBlocks{2}.PeakingGain = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25];
rxBlocks{3}.ConfigSelect = Ctle;% Here we will use Ctle variable to sweep CTLE vector
rxBlocks{3}.Specification = 'DC Gain and Peaking Gain';
rxBlocks{3}.PeakingFrequency = 32000000000;
rxBlocks{3}.DCGain = [0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -10 -11 -12 -13 -14 -15 -16 -17 -18 -19 -20 -
21 -22 -23 -24 -25];
rxBlocks{3}.PeakingGain = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25];
rxBlocks{4} = serdes.DFECDR;
rxBlocks{4}.BlockName = 'DFECDR';
rxBlocks{4}.Mode = 2;
rxBlocks{4}.TapWeights = [0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0];
rxBlocks{4}.MinimumTap = -.7;
rxBlocks{4}.MaximumTap = .7;
% ------------------------------------
% Build txModel:
txAnalogModel = AnalogModel( ...
    'R',50, ...
    'C',1.000000e-13);
tx = Transmitter( ...
    'Blocks',txBlocks, ...
    'AnalogModel',txAnalogModel, ...
    'RiseTime',1.000000e-11, ...
    'VoltageSwingIdeal',1, ...
    'Name','TX');
% ------------------------------------
% Build rxModel:
rxAnalogModel = AnalogModel( ...
    'R',50, ...
    'C',2.000000e-13);
rx = Receiver( ...
    'Blocks',rxBlocks, ...
    'AnalogModel',rxAnalogModel, ...
    'Name','RX');
% ------------------------------------
% Channel definition
   channel = ChannelData( ...
    'ChannelLossdB',ChannelLoss, ...
    'ChannelLossFreq',32000000000, ...
    'ChannelDifferentialImpedance',100);
% ------------------------------------
% Assembling system
PAM4_systems = PAM4_sys(tx,rx,channel,1e-10,16,4,1e-06);

% Getting system results
serdesResults = analysis(PAM4_systems);

%----------------- Eye Calculation without jitter -----------
%
%Calculate Statistical Eye
nwaves = size(serdesResults.pulse,2)/2;

[stateye,vh,th] = pulse2stateye(serdesResults.pulse(:,nwaves+1:end),...
    SamplesPerSymbol,ModulationLevels);
[~,prefixstr2,Y2] = serdes.utilities.num2prefix(SymbolTime);
th2 = th*SymbolTime*Y2;
% ------------------------------------
%Calculate Eye Metrics
 [~,~,contours,~,EH,~,~,~,~,~,EW,~,~,~,~,eyeAreas,~,COM] = ...
          serdes.utilities.calculatePAMnEye(ModulationLevels,BERtarget, ...
     th2(1),th2(length(th2)),vh(1),vh(length(vh)),stateye);
% ------------------------------------
% Eye Diagram Measurements
% EH_noJitter = EH
% EW_noJitter = EW
% COM_noJitter = COM

%--------------------- Adding Jitter ----------------
%
% Build Jitter And Noise Object:
jitter = JitterAndNoise( ...
    'Tx_Dj',Tx_Dj,...
    'Tx_Sj',Tx_Sj,...
    'Rx_Dj',Rx_Dj,...
    'Rx_Sj',Rx_Sj,...
    'RxClockMode','clocked');
% ------------------------------------
% Assembling system and getting system results
PAM4_systems_Jitter = PAM4_sys_Jitter(tx,rx,channel,jitter,1e-10,16,4,1e-06);

vh = PAM4_systems_Jitter.Eye.Vh;
th = PAM4_systems_Jitter.Eye.Th;
stateye = PAM4_systems_Jitter.Eye.Stateye;

%----------------- EyeCalculation with jitter -----------
%
%Calculate Statistical Eye

th2 = th*SymbolTime*Y2;

% ------------------------------------
%Calculate Eye Metrics with Jitter

 [~,~,contours,~,EH,~,~,~,~,~,EW,~,~,~,~,eyeAreas,~,COM] = ...
          serdes.utilities.calculatePAMnEye(ModulationLevels,BERtarget, ...
     th2(1),th2(length(th2)),vh(1),vh(length(vh)),stateye);

% ------------------------------------
% Eye Diagram Measurements  with Jitter
EHj = EH;
```

```
EWj = EW;
% COM_withJitter = COM

% The objective function is COM. It will be negative as we are looking to
% maximize COM with fminsearch
% The channel operating margin (COM) is a ratio between the signal and the noise and is given by
the equation:
% COM=20log_10(Signal/Noise), where:
% Signal = signal amplitude from the pulse response cursor voltage (the mean eye level height).
% Noise = The noise amplitude is estimated at a given BER by the sum of intersymbol interference
(ISI) voltages.

 u = -COM;

end
```

PAM4 optimization function

```
% PAM4_Fun_Opt.m
%==================================================
% PCIeGen6 - PAM4 Equalizers Optimization
% Francisco E. Rangel-Patino and Roberto J. Ruiz-Urbina
% Research Group on Computer-Aided Engineering of Circuits and Systems
% (CAECAS)
% Department of Electronics, Systems, and Informatics ITESO
% Intel Guadalajara Design Center - Electrical Validation
% January 2021
% Version V12
%==================================================
%                 DESIGN VARIABLES
%%% Define CTLE and FIR Filter Coefficients Vectors per PCIeGen6 Spec
% Cm2 = 1/24
% x1: CTLE = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
% x2: Cm1 = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24]
% x3: Cp  = -1*[0/24 1/24 2/24 3/24 4/24 5/24 6/24 7/24 8/24]
%     C0  = 1 - |Cm1| - |Cp| - |Cm2|
% Define TapWeights Vector for SerDes ToolBox Simulator
% txBlocks{1}.TapWeights = [0 Cm2 Cm1 C0 Cp]
%==================================================
function U=
PAM4_Fun_Opt(Xo,u0,Cm2,Cm1V,CpV,CTLEV,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BE
Rtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj)
global fevaluations U_i ctle_i Cm1_i Cp_i

fevaluations = fevaluations + 1;

% Seed: Xo = [CTLE, Cm1, Cp]
% Build cell array of Tx blocks:
% Defining limits for Xo values based on CTLE, Cm1, and Cp vectors
if Xo(1) < CTLEV(1)
        Xo(1) = CTLEV(1);
    elseif Xo(1) > CTLEV(26)
        Xo(1) = CTLEV(26);
end
if Xo(2) > Cm1V(1)
        Xo(2) = Cm1V(1);
    elseif Xo(2) < Cm1V(7)
        Xo(2) = Cm1V(7);
end
 if Xo(3) > CpV(1)
        Xo(3) = CpV(1);
    elseif Xo(3) < CpV(9)
        Xo(3) = CpV(9);
 end

  if Xo(2)==0/24 && Xo(3)<-8/24
     U = 10000;
      return
  end
 if Xo(2)==-1/24 && Xo(3)<-7/24
     U = 10000;
      return
 end

 if Xo(2)==-2/24 && Xo(3)<-6/24
     U = 10000;
      return
 end

 if Xo(2)==-3/24 && Xo(3)<-5/24
     U = 10000;
      return
 end

 if Xo(2)==-4/24 && Xo(3)<-4/24
     U = 10000;
      return
 end

 if Xo(2)==-5/24 && Xo(3)<-3/24
     U = 10000;
      return
 end

 if Xo(2)==-6/24 && Xo(3)<-2/24
     U = 10000;
      return
 end

Ctle = round(Xo(1));% CTLE Gain Value
Cm1 = Xo(2);
Cp = Xo(3);
C0 = 1 - abs(Cm1) - abs(Cp) - abs(Cm2);

if C0 < 0.625
      U = 10000;
      return
end
%
% Compute "u" map center
[u,EHj,EWj] =
PAM4_SerDes(Cm2,Cm1,Cp,Ctle,C0,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BERtarget
,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
uc=u;
%
% Compute "u" up map
Cm1_up = (Cm1*24+1)/24;
C0_up = 1 - abs(Cm1_up) - abs(Cp) - abs(Cm2);
[u,EHj,EWj] =
PAM4_SerDes(Cm2,Cm1_up,Cp,Ctle,C0_up,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,BER
target,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
uy_p=u;
%
% Compute "u" down map
Cm1_down = (Cm1*24-1)/24;
C0_down = 1 - abs(Cm1_down) - abs(Cp) - abs(Cm2);
[u,EHj,EWj] =
PAM4_SerDes(Cm2,Cm1_down,Cp,Ctle,C0_down,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime
,BERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
uy_n=u;
%
% Compute "u" left map
Cp_left = (Cp*24-1)/24;
C0_left = 1 - abs(Cm1) - abs(Cp_left) - abs(Cm2);
[u,EHj,EWj] =
PAM4_SerDes(Cm2,Cm1,Cp_left,Ctle,C0_left,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime
,BERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
```

```
ux_n=u;
%
% Compute "u" right map
Cp_right = (Cp*24+1)/24;
C0_right = 1 - abs(Cm1) - abs(Cp_right) - abs(Cm2);
[u,EHj,EWj] =
PAM4_SerDes(Cm2,Cm1,Cp_right,Ctle,C0_right,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTi
me,BERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
ux_p=u;
%
% Compute U(x)
l1 = abs(0.8*uc)-abs(uy_p);
l2 = abs(0.8*uc)-abs(uy_n);
l3 = abs(0.8*uc)-abs(ux_p);
l4 = abs(0.8*uc)-abs(ux_n);
L = [0,l1,l2,l3,l4]; %Define penalty vector
gamma = abs(u0)/(norm(max(L)))^2; % Compute penalty coefficient
U = uc + gamma*(norm(L))^2; % Compute and display Objective
U_i(fevaluations) = U;
ctle_i(fevaluations) = Ctle;
Cm1_i(fevaluations) = Cm1;
Cp_i(fevaluations) =  Cp;
End
```

PAM4 sys function

```
function[sys] = PAM4_sys(tx,rx,channel,~,~,~,~)

SymbolTime = 31e-12;% PCIe6 UI
SamplesPerSymbol = 64;
ModulationLevels = 4;
BERtarget = 1e-06;
sys = SerdesSystem(...
    'TxModel',tx,...
    'RxModel',rx,...
    'ChannelData',channel,...
    'SymbolTime',SymbolTime, ...
    'SamplesPerSymbol',SamplesPerSymbol, ...
    'Modulation',ModulationLevels, ...
    'Signaling','Differential', ...
    'BERtarget',BERtarget);
```

PAM4 sys jitter function

```
function[sys] = PAM4_sys_Jitter(tx,rx,channel,jitter,~,~,~,~)

SymbolTime = 31e-12;% PCIe6 UI
SamplesPerSymbol = 16;
ModulationLevels = 4;
BERtarget = 1e-06;
sys = SerdesSystem(...
    'TxModel',tx,...
    'RxModel',rx,...
    'ChannelData',channel,...
    'JitterAndNoise',jitter,...
    'SymbolTime',SymbolTime, ...
    'SamplesPerSymbol',SamplesPerSymbol, ...
    'Modulation',ModulationLevels, ...
    'Signaling','Differential', ...
    'BERtarget',BERtarget);
```

PAM4 plot function

```
function[EH] = PAM4_Plot(serdesResults,SymbolTime,SamplesPerSymbol,ModulationLevels,BERtarget)

%Visualize Pulse Response
[~,prefixstr1,Y1] = serdes.utilities.num2prefix(SymbolTime*127);
```

```
numberOfWaves = size(serdesResults.impulse,2)/2;
figure
plot(serdesResults.t1*Y1,serdesResults.pulse)
xlabel("["+prefixstr1+"]")
ylabel('[V]')
grid on
legendCell = cell(numberOfWaves,2);
for ii = 1:numberOfWaves
    legendCell{ii,1} = sprintf('Unequalized p_%i(t)',ii-1);
    legendCell{ii,2} = sprintf('Equalized p_%i(t)',ii-1);
end
legend(legendCell(:));
title('Pulse Response')

%Visualize PRBS Waveform Response
figure
plot(serdesResults.t2*Y1,serdesResults.wave)
xlabel("["+prefixstr1+"]")
ylabel('[V]')
grid on
legendCell = cell(numberOfWaves,2);
for ii = 1:numberOfWaves
    legendCell{ii,1} = sprintf('Unequalized w_%i(t)',ii-1);
    legendCell{ii,2} = sprintf('Equalized w_%i(t)',ii-1);
end
legend(legendCell(:));
title('PRBS Waveform')

%Calculate Statistical Eye
nwaves = size(serdesResults.pulse,2)/2;

[stateye,vh,th] = pulse2stateye(serdesResults.pulse(:,nwaves+1:end),...
    SamplesPerSymbol,ModulationLevels);
[~,prefixstr2,Y2] = serdes.utilities.num2prefix(SymbolTime);
th2 = th*SymbolTime*Y2;

%Calculate Eye Metrics
[~,~,contours,bathtubs,EH,~,~,~,~,~,~,~,EW] = ...
    serdes.utilities.calculatePAMnEye(ModulationLevels,BERtarget, ...
    th2(1),th2(length(th2)),vh(1),vh(length(vh)),stateye);

si_eyecmap = serdes.utilities.SignalIntegrityColorMap;
linecolor = [0.75 0 0.75];

%Plot bathtubs, Statistical Eye and contours
figure
title('Statistical Eye')
yyaxis('right')
ylabel('Probability')
semilogy(th2,10.^bathtubs,'color',linecolor,'linewidth',2,'linestyle','-')
set(gca,'YColor',linecolor)

yyaxis('left')
hold('on')
imagesc(th2,vh,stateye)
axis('xy');
colormap(si_eyecmap)

plot(th2,contours,'m-','linewidth',2)
xlabel("["+prefixstr2+"]")
ylabel('[V]')

%Display Report
fprintf('\nSerdes Analysis Summary Report\n');
switch ModulationLevels
    case 4
        eyeLabel = {'Lower','Center','Upper'};
        for ii = length(EH):-1:1
            fprintf('Eye Height %s (V)  %g\n',eyeLabel{ii},EH(ii))
        end
        for ii = length(EW):-1:1
            fprintf('Eye Width %s (%s)  %g\n',eyeLabel{ii},prefixstr2,EW(ii))
        end
    case 2
        fprintf('Eye Height (V) %g\n',EH)
        fprintf('Eye Width (%s) %g\n',prefixstr2,EW)
end

for ii = 1:length(serdesResults.outparams)
    if ~isempty(serdesResults.outparams{ii})
        sout = serdes.utilities.FlattenStruct(serdesResults.outparams{ii});
        for jj = 1:size(sout,1)
            fprintf('%s     %s\n',sout{jj,1:2})
        end
    end
end
```

PAM4 eq map function

```
% PAM4_EQmap.m
%==================================================
% PCIeGen6 - PAM4 Equalizers Optimization
% Francisco E. Rangel-Patino and Roberto J. Ruiz-Urbina
% Research Group on Computer-Aided Engineering of Circuits and Systems
% (CAECAS)
% Department of Electronics, Systems, and Informatics ITESO
% Intel Guadalajara Design Center - Electrical Validation
% January 2021
% Version V01
%==================================================
% management functions
clear  % clear all variable/information in the workspace
clear global  %  again use caution - clears global information
clc    % position the cursor at the top of the screen
format compact  % avoid skipping a line when writing to the command window
warning off  % don't report any warnings like divide by zero etc
%==================================================
%                 DESIGN VARIABLES
%%% Define CTLE and FIR Filter Coefficients Vectors per PCIeGen6 Spec
% Cm2 = 1/24
% x1: CTLE = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
% x2: Cm1 = [0/24 1/24 2/24 3/24 4/24 5/24 6/24]
% x3: Cp  = [0/24 1/24 2/24 3/24 4/24 5/24 6/24 7/24 8/24]
%     C0  = 1 - Cm1 - Cp - Cm2
% Define TapWeights Vector for SerDes ToolBox Simulator
% txBlocks{1}.TapWeights = [0 Cm2 Cm1 C0 Cp]
%==================================================
%                 SERDES SYSTEM SETUP
%<We will use integers numbers for Cm, Cp and CTLE values for optimization, but will convert to
actual values (x/24) for simulation>
Cm2  = 1/24; % Tx FIR Filter coefficient Cm2
Cm1V  = -1*[0 1 2 3 4 5 6]/24; % Tx FIR Filter coefficient Cm1
CpV   = -1*[0 1 2 3 4 5 6 7 8]/24; % Tx FIR Filter coefficient Cp
Ctle = 8;
%CTLEV = [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]; % Rx CTLE values
% ChannelLoss: Channel losses value
ChannelLoss = 25;
% SamplesPerSymbol: Number of data points per symbol.
% The Samples per symbol determine the acquisition bandwidth. PCIeGen6 Data Rate (Gb/s) = 64.0
SamplesPerSymbol = 64;
%  ModulationLevels: Number of logic levels in the modulation scheme: PAM4
ModulationLevels = 4;
% SymbolTime: Time it takes to send one symbol across the link.
% Per PCIeBase Specification for  PCIe Gen6: Unit Interval
% (UI(Tx))=31.246875 psec
SymbolTime = 31e-12;
% BERtarge: Target bit error rate. Target bit error rate used to generate eye-contours,
% specified as a unitless real positive scalar. PAM4 requires higher BER at the physical layer
(~1e-6)
BERtarget = 1e-06;
% Jitter Configuration: Jitter parameters defined as Type UI:
Tx_Dj = .5e-12; % Tx Deterministic jitter
Tx_Sj = .3e-12; % Tx sinusoidal jitter
Rx_Dj = .4e-12; % Rx Deterministic jitter
Rx_Sj = .7e-12; % Rx sinusoidal jitter
%==================================================
%                   COM Equalization Map

 for C = 1:length(CpV)

       for R = 1:length(Cm1V)
           fprintf('\Running EQ Map\n')
           C0 = 1 - abs(Cm1V(R)) - abs(CpV(C)) - abs(Cm2);
           COM_EQmap(R,C)=
PAM4_SerDes(Cm2,Cm1V(R),CpV(C),Ctle,C0,ChannelLoss,SamplesPerSymbol,ModulationLevels,SymbolTime,B
ERtarget,Tx_Dj,Tx_Sj,Rx_Dj,Rx_Sj);
        end
 end

figure
contourf(CpV,Cm1V,COM_EQmap,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                      % add axis labels and plot title
ylabel('Cm1');
title('COM EQ Map');

figure
surf(CpV,Cm1V,COM_EQmap)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                      % add axis labels and plot title
ylabel('Cm1');
title('COM EQ Map');
```

Eq map contour plot

```
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Francisco Rangel November 20, 2014
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% SunrisePoint PCIe3 Equalization MAPS(SPT_PCIe3_EQmaps.m)
% Platform: RVP7 (GDC301724)
% Silicon: GDC_M107226
% Device: Broadcom
% Port: 9 (4 Lanes)
% Channels: L12=8.1", L13=7.7", L14=8.5", L15=8.2"
% DATA: Margin sweeping EQ Coefficients Cp and Cm
% Analysis: Coefficients selection base on Minimum of "Min_Eye" and Maximum
% of "Eye_Sym". Plot EQ Maps of and then compute EQ Maps average.
% BIOS: Version BIOS 56_95 with register mm_phs_off=0x3. Equivalent to BIOS
% 56_74 for EQ Maps.
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Matrix of Coefficients
Cp = [0 1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20];
Cm = [2 4 6 8 10 12 14 16]';
%
Area = [  9 8   7   7   5   6   6   5   6   7   5   5   6   4   5   5   5   5   6   5   5;
                   9    8   6   7   6   6   6   5   6   6   5   5   6   6   5   5   5   5   5   0   0;
                   10   8   5   8   5   6   6   5   6   4   5   5   6   6   6   5   4   0   0   0   0;
                   7    10  6   7   6   5   4   6   5   4   7   7   6   6   5   0   0   0   0   0   0;
```

```
                   8   7   6   6   4   6   4   5   4   6   6   5   6   0   0   0   0   0   0   0   0;
                   7   8   5   6   6   6   6   5   6   5   5   0   0   0   0   0   0   0   0   0   0;
                   8   7   4   6   5   4   5   2   4   0   0   0   0   0   0   0   0   0   0   0   0;
                   4   4   1   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0];
% Maximum of EYE_SYM for Lane12
EQ_L12_EYEsym = [   2   5   1   4   3   3   3   3   3   1   3   2   1   3   2   2   2   2   0   2   4;
                    3   2   2   2   3   4   4   4   5   4   5   6   4   3   5   3   1   1   2   0   0;
                    3   2   3   2   4   4   4   4   3   5   5   5   3   2   2   2   3   0   0   0   0;
                    2   1   2   1   2   2   3   1   2   3   2   1   1   2   2   0   0   0   0   0   0;
                    1   3   2   2   3   1   3   3   3   1   2   2   1   0   0   0   0   0   0   0   0;
                    2   2   2   1   2   1   2   2   2   3   2   0   0   0   0   0   0   0   0   0   0;
                    2   1   3   2   2   3   3   4   4   0   0   0   0   0   0   0   0   0   0   0   0;
                    5   5   7   5   6   5   6   0   0   0   0   0   0   0   0   0   0   0   0   0   0];
figure(1)
%contourf(Cp,Cm,EQ_L12_MinEYE,20)
%colormap(jet(100))
%axis tight
%grid on
%colorbar
%surfl(Cp,Cm,EQ_L12_MinEYE)
%colormap(jet(100))    % change color map
%shading interp    % interpolate colors across lines and faces
%xlabel('Cp');                        % add axis labels and plot title
%ylabel('Cm');
%title('Minimum of MinEye Port9 Lane12 (8.1") Broadcom');
%plot(EQ_L12_MinEYE,EQ_L12_EYEsym,'ok');

figure(2)
%contourf(Cp,Cm,EQ_L12_MinEYE,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                        % add axis labels and plot title
ylabel('Cm');
title('Minimum of MinEye Port9 Lane12 (8.1") Broadcom');

%-------------------------------------------------------
% Minimum of Min_EYE for Lane13
EQ_L13_MinEYE = [   7   8   4   5   5   4   4   3   2   4   3   3   1   2   1   1   1   1   1   0   2;
                    8   6   5   5   4   3   5   4   3   3   3   3   3   2   2   2   2   2   3   0   0;
                    7   7   6   6   6   5   6   4   4   5   5   5   5   5   4   4   4   0   0   0   0;
                    7   5   4   3   5   3   4   3   3   3   2   3   3   3   4   0   0   0   0   0   0;
                    4   5   3   3   2   3   3   2   4   3   3   3   2   0   0   0   0   0   0   0   0;
                    3   3   3   3   2   2   3   3   2   2   1   0   0   0   0   0   0   0   0   0   0;
                    3   3   2   2   3   2   2   0   1   0   0   0   0   0   0   0   0   0   0   0   0;
                    2   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

% Maximum of EYE_SYM for Lane13
EQ_L13_EYEsym = [   4   3   3   2   2   3   3   3   6   5   6   5   7   7   9   6   6   6   5   9   3;
                    1   2   2   3   4   5   3   4   4   4   4   4   5   5   5   4   3   4   4   0   0;
                    2   2   1   0   2   1   2   3   2   1   1   0   1   1   2   1   1   0   0   0   0;
                    2   2   4   4   4   5   4   5   4   5   4   4   2   2   1   0   0   0   0   0   0;
                    4   5   4   5   6   4   5   6   3   3   2   2   3   0   0   0   0   0   0   0   0;
                    4   4   5   4   5   3   4   3   4   3   3   0   0   0   0   0   0   0   0   0   0;
                    5   5   4   4   3   3   3   5   4   0   0   0   0   0   0   0   0   0   0   0   0;
                    4   5   4   3   4   0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

figure(3)
plot(EQ_L13_MinEYE,EQ_L13_EYEsym,'ok');
%contourf(Cp,Cm,EQ_L13_MinEYE,20)
%colormap(jet(100))
%axis tight
%grid on
%colorbar
%xlabel('Cp');                        % add axis labels and plot title
%ylabel('Cm');
%title('Minimum of MinEye Port9 Lane13 (7.7") Broadcom');

figure(4)
contourf(Cp,Cm,EQ_L13_EYEsym,20)
colormap(jet(100))
```

```
axis tight
grid on
colorbar
xlabel('Cp');                      % add axis labels and plot title
ylabel('Cm');
title('Maximum of MinSymetry Port9 Lane13 (7.7") Broadcom');

%------------------------------------------------------
% Minimum of Min_EYE for Lane14
EQ_L14_MinEYE = [   11  11  11  9   10  8   8   9   9   8   6   8   7   5   6   4   4   3   4   2   4;
                    12  13  8   8   8   9   9   9   8   7   7   6   6   5   5   5   4   4   4   0   0;
                    13  13  7   9   10  8   9   8   7   6   6   6   6   4   4   4   4   0   0   0   0;
                    9   10  8   7   6   7   7   7   6   6   5   5   5   5   5   0   0   0   0   0   0;
                    8   10  8   7   6   7   8   6   5   5   5   5   5   0   0   0   0   0   0   0   0;
                    8   8   8   7   7   6   7   6   4   5   5   0   0   0   0   0   0   0   0   0   0;
                    8   8   5   6   6   6   6   4   4   0   0   0   0   0   0   0   0   0   0   0   0;
                    3   3   2   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

% Maximum of EYE_SYM for Lane14
EQ_L14_EYEsym = [   4   3   3   5   4   5   7   4   1   5   3   1   2   5   2   3   4   3   3   5   6;
                    4   3   9   6   7   2   4   1   1   2   2   5   5   6   6   5   6   5   8   0   0;
                    3   1   3   3   2   4   1   2   4   9   9   8   3   4   8   6   7   0   0   0   0;
                    4   4   4   5   5   3   4   3   3   3   5   5   4   4   4   0   0   0   0   0   0;
                    4   3   2   2   2   2   1   2   2   3   3   3   3   0   0   0   0   0   0   0   0;
                    2   2   2   2   2   1   0   1   2   2   2   0   0   0   0   0   0   0   0   0   0;
                    1   0   3   2   3   1   1   1   1   0   0   0   0   0   0   0   0   0   0   0   0;
                    6   7   5   5   3   1   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

figure(5)
plot(EQ_L14_MinEYE,EQ_L14_EYEsym,'ok');
%contourf(Cp,Cm,EQ_L14_MinEYE,20)
%colormap(jet(100))
%axis tight
%grid on
%colorbar
%xlabel('Cp');                      % add axis labels and plot title
%ylabel('Cm');
%title('Minimum of MinEye Port9 Lane14 (8.5") Broadcom');

figure(6)
contourf(Cp,Cm,EQ_L14_EYEsym,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                      % add axis labels and plot title
ylabel('Cm');
title('Maximum of MinSymetry Port9 Lane14 (8.5") Broadcom');

%------------------------------------------------------
% Minimum of Min_EYE for Lane15
EQ_L15_MinEYE = [   9   10  9   10  10  8   9   8   7   7   5   6   6   5   5   4   4   4   3   3   3;
                    12  11  9   9   8   6   8   7   6   5   5   6   5   4   4   4   4   3   4   0   0;
                    10  10  9   9   8   8   7   7   6   7   7   7   5   6   6   6   5   0   0   0   0;
                    9   9   8   8   7   6   7   6   6   5   5   5   5   5   4   0   0   0   0   0   0;
                    8   8   6   6   5   5   5   5   4   4   4   4   4   0   0   0   0   0   0   0   0;
                    7   5   3   4   4   4   3   4   4   2   1   0   0   0   0   0   0   0   0   0   0;
                    5   4   3   3   3   1   2   3   2   0   0   0   0   0   0   0   0   0   0   0   0;
                    2   2   0   1   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

% Maximum of EYE_SYM for Lane15
EQ_L15_EYEsym = [   4   5   7   5   3   7   5   7   12  9   14  11  12  12  14  13  11  14  15  13  13;
                    1   1   4   6   8   10  8   9   9   11  10  9   9   10  11  8   8   9   8   0   0;
                    3   4   6   4   6   4   6   5   7   3   5   5   6   4   5   4   5   0   0   0   0;
                    3   4   5   4   7   8   7   7   8   9   8   9   8   8   9   0   0   0   0   0   0;
                    8   6   10  10  10  10  11  11  12  12  12  11  11  0   0   0   0   0   0   0   0;
                    10  11  14  13  13  14  15  12  12  15  15  0   0   0   0   0   0   0   0   0   0;
                    14  14  16  15  16  15  15  14  12  0   0   0   0   0   0   0   0   0   0   0   0;
                    17  15  13  15  11  0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0];

figure(7)
plot(EQ_L15_MinEYE,EQ_L15_EYEsym,'ok');
%contourf(Cp,Cm,EQ_L15_MinEYE,20)
```

```
%colormap(jet(100))
%axis tight
%grid on
%colorbar
%xlabel('Cp');                       % add axis labels and plot title
%ylabel('Cm');
%title('Minimum of MinEye Port9 Lane15 (8.2") Broadcom');

figure(8)
contourf(Cp,Cm,EQ_L15_EYEsym,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                       % add axis labels and plot title
ylabel('Cm');
title('Maximum of MinSymetry Port9 Lane15 (8.2") Broadcom');

%--------------------------------------------------------
% AVERGAGE of Minimums (Min Eye) & AVERAGE of Maximums (Eye Symetry)
%MEAN_EQ_MinEYE=(EQ_L12_MinEYE+EQ_L13_MinEYE+EQ_L14_MinEYE+EQ_L15_MinEYE)/4;
MEAN_EQ_EYEsym=(EQ_L12_EYEsym+EQ_L13_EYEsym+EQ_L14_EYEsym+EQ_L15_EYEsym)/4;

figure(9)
%contourf(Cp,Cm,MEAN_EQ_MinEYE,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                       % add axis labels and plot title
ylabel('Cm');
title('Mean of Minimum of MinEye Port9 Lanes 12,13,14,15 Broadcom');

figure(10)
contourf(Cp,Cm,MEAN_EQ_EYEsym,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                       % add axis labels and plot title
ylabel('Cm');
title('Mean Maximum of MinSymetry Port9 Lane 12,13,14,15 Broadcom');

%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
% Platform: RVP7 (GDC301732)
% Silicon: GDC_M107230
% Device: Broadcom
% Port: 4 (x1 Lanes)
% Channels: L7=9.5"
% BIOS: 56_74
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Cp2 = [0 2 4 6 8 10];
Cm2 = [2 4 6 8 10 12]';
% Minimum of Min_EYE for Lane7
EQ_L7_MinEYE = [  1 0    0   0   1   1;
                  4 1    2   2   2   3;
                  3 2    3   3   2   2;
                  5 4    4   4   5   5;
                  6 5    4   5   4   3;
                  4 4    3   3   2   2];

% Maximum of EYE_SYM for Lane7
EQ_L7_EYEsym = [  15     0   13  9   10  9;
                    8    10  9   8   9   6;
                    9    7   6   7   6   8;
                    0    3   3   2   1   0;
                    1    0   0   1   3   5;
                    1    4   3   3   6   8];
```

```
figure(11)
plot(EQ_L7_MinEYE,EQ_L7_EYEsym,'ok');
%contourf(Cp2,Cm2,EQ_L7_MinEYE,20)
%colormap(jet(100))
%axis tight
%grid on
%colorbar
%xlabel('Cp');                      % add axis labels and plot title
%ylabel('Cm');
%title('Minimum of MinEye Port4 Lane7 (9.5") Broadcom');

figure(12)
contourf(Cp2,Cm2,EQ_L7_EYEsym,20)
colormap(jet(100))
axis tight
grid on
colorbar
xlabel('Cp');                      % add axis labels and plot title
ylabel('Cm');
title('Maximum of MinSymetry Port4 Lane7 (9.5") Broadcom');

%------------------------------------------------------
```

Knob function

```
function [knob] = knobfilter(s,x)

[~,ii] = min(bsxfun(@(x,y)abs(x-y),s(:).',x(:)),[],2);
knob = s(ii);

end
```

References

[1] PCI SIG Org. (2020), *PCI Express® Base Specification Revision 6.0 Version 0.7* [Online]. Available: https://pcisig.com/specifications.

[2] F. E. Rangel-Patiño, J. E. Rayas-Sánchez, E. A. Vega-Ochoa, and N. Hakim, "Direct optimization of a PCI Express link equalization in industrial post-silicon validation," in *IEEE Latin American Test Symp. (LATS 2018)*, Sao Paulo, Brazil, Mar. 2018, pp. 1-6.

[3] M. Jackson, R. Budruk, *PCI Express Technology – Comprehensive Guide to Generations 1.x, 2.x, and 3.0*, MindShare, Inc., 2012.

[4] D. Das-Sharma, "PCIe® 6.0 Specification: The Interconnect for I/O Needs of the Future," *PCI-SIG® Educational Webinar Series*, June 2020. [Online]. Available: https://pcisig.com/

[5] R. Solomon, "Life in the Fast Lane: PCI Express® Technology in Automotive". [Online]. Available: https://pcisig.com/

[6] D. Das-Sharma, "What's the Difference from PCIe 3.0 to PCIe 6.0?," *Electronic Design*, Jul. 2020. [Online]. Available: http://www.electronicdesign.com

[7] Q. Liao et al., "The Design Techniques for High-Speed PAM4 Clock and Data Recovery," in *IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA 2018)*, Beijing, China, 2018, pp. 142-143.

[8] J. He, *High Speed Serial Link design with multi-level signaling and characteristic impedance extraction from a transmission line with meshed ground planes*. Masters Thesis, Electrical and Computer Engineering Commons Department, Missouri University of Science and Technology, Missouri, USA, 2017.

[9] R. J. Ruiz-Urbina, F. E. Rangel-Patiño, O. H. Longoria-Gandara, J. E. Rayas-Sánchez and E. A. Vega-Ochoa "Transmitter and Receiver Equalizers Optimization for PCI Express Gen 6.0 based on PAM4," in *IEEE MTT-S Latin America Microwave Conf. (LAMC-2021)*, Cali, Colombia, May 2021.

[10] Application Note *AN835: PAM4 Signaling Fundamentals* (2019)_[Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/an/an835.pdf

[11] Keysight Technologies, Application Note *PAM-4 Design Challenges and the implications Test* [Online]. Available: https://www.keysight.com/zz/en/assets/7018-04746/application-notes/5992- 0527.pdf

[12] G. Zhang, M. Huang, H. Zhang, *COM for PAM4 Link Analysis – what you need to know*, [Online]. Available: https://www.edicononline.com/wp-content/uploads/sites/2/2019/10/1_COM-for-PAM4-Link-Analysis.pdf

[13] N. Dikhaminjia, J. He, H. Deng, M. Tsiklauri, and J. Drewniak "Effect of improved optimization of DFE equalization on crosstalk and jitter in high speed links with multi-level signal," in *IEEE 68th Electronic Components and Technology Conference (ECTC 2018),* San Diego, CA, USA, 2018.

[14] J. He, N. Dikhaminjia, M. Tsiklauri, J. Drewniak, A. Chada, and B. Mutnury, "Equalization enhancement approaches for PAM4 signaling for next generation speeds," in *IEEE 67th Electronic Components and Technology Conf. (ECTC 2017)*, Orlando, FL, 2017, pp. 1874-1879.

[15] Viveros-Wacher and J. E. Rayas-Sánchez, “Eye diagram optimization based on design of experiments (DoE) to accelerate industrial testing of high speed links,” in *IEEE MTT-S Latin America Microwave Conf. (LAMC-2016)*, Puerto Vallarta, Mexico, Dec. 2016, pp. 1-3.

[16] P. Kuma Hanumolu, G. Wei, and U. Moon, “Equalizers for high-speed serial links,” International Journal of High-Speed Electronics and Systems, vol. 15, no. 2, pp. 429-458, 2005

[17] F. E. Rangel-Patiño, J. E. Rayas-Sánchez, and N. Hakim, “Transmitter and receiver equalizers optimization methodologies for high-speed links in industrial computer platforms post-silicon validation” in *IEEE International Test Conference (ITC-2018)*, Phoenix, AZ, USA

[18] F. E. Rangel-Patiño, J. L. Chávez-Hurtado, A. Viveros-Wacher, J. E Hakim, “System margining surrogate-based optimization in post-silico Microwave Theory Techn., vol. 65, no. 9, pp. 3109-3115, Sep. 2017

[19] F. E. Rangel-Patiño, A. Viveros-Wacher, J. E. Rayas-Sánchez, E. A. Vega-Ochoa, I. Duron-Rosales, and N. Hakim, “A holistic methodology for system margining and jitter tolerance optimization in post-silicon validation,” in IEEE MTT-S Latin America Microwave Conf. (LAMC-2016), Puerto Vallarta, Mexico, Dec. 2016, pp. 1-4

[20] B. Gore, and R. Mellitz, “An exercise in applying channel operating margin (COM) for 10GBASE-KR channel design,” in IEEE International Symposium on Electromagnetic Compatibility (EMC-2014), Raleigh, NC, USA

[21] PCI SIG Org. (2020), *PCI Express® Base Specification Revision 6.0 Version 0.7* [Online]. Available: https://pcisig.com/specifications.

[22] F. de Paulis, T. Wang-Lee, R. Mellitz, M. Resso, R. Rabinovich, and O. J. Danzy, “Backplane channel design exploration at 112 Gbps using channel operating margin (COM),” in *IEEE Int. Symp. Electromagnetic Compatibility & Signal/Power Integrity (EMCSI 2020)*, Reno, NV, USA, 2020, pp. 158-163.

[23] “IEEE Standard for Ethernet Amendment 2: Physical Layer Specifications and Management Parameters for 100 Gb/s Operation Over Backplanes and Copper Cables,” in *IEEE Std 802.3bj-2014 (Amendment to IEEE Std 802.3-2012 as amended by IEEE Std 802.3bk-2013)* , vol., no., pp.1-368, 3 Sept. 2014, doi: 10.1109/IEEESTD.2014.6891095

[24] Anritsu, *PAM4 Gigabit Ethernet Electrical SERDES Analysis, Debug and Compliance Testing* [Online]. Available: www.anritsu.com

[25] Matlab SerDes Toolbox, https://www.mathworks.com/products/serdes.html

[26] F. E. Rangel-Patiño, J. E. Rayas-Sánchez, A. Viveros-Wacher, J. L. Chávez-Hurtado, E. A. Vega-Ochoa, and N. Hakim “Post-Silicon Receiver Equalization Metamodeling by Artificial Neural Networks” *IEEE Trans. Computer-Aided Design of integrated Circuits and Systems,* Vol. 38, no. 4, pp. 733-740, Apr. 2019

[27] F. E. Rangel-Patiño, A. Viveros-Wacher, J. E. Rayas-Sánchez, I. Duron-Rosales, E. A. Vega-Ochoa, N. Hakim and E. Lopez-Miralrio, “A Holistic Formulation for System Margining and Jitter Tolerance Optimization in Industrial Post-Silicon Validation’ *IEEE Trans. Emerging Topics Computing*, vol. 8 no. 2 pp. 453-463, Apr.-Jun. 2020.

[28] Matlab, *CEI-56G-LR Transmitter/Receiver IBIS-AMI Model* [Online]. Available: https://www.mathworks.com/help/serdes/ug/cei-56g-lr-transmitter-receiver-ibis-ami-model.html

Subject Index