DISSERTATION

submitted

to the

Combined Faculty for the Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Dipl.Phys. Dipl.Inf. Maximilian Thürmer

Born in Heidelberg on August 31st, 1983

Date of oral examination: _____

Modelling and performance analysis of multigigabit serial interconnects using real number based analog verification methods

Advisor: Prof. Dr. Ulrich Brüning

Danksagungen

Wissenschaftlich gesehen mögen wir alle auf den Schultern von Giganten stehen, doch selbst dieser aufrechte Stand ist uns nur möglich, dank der uns liebenden Menschen, die sich um uns sorgen und Worte der Empathie für uns in den Phasen finden, in denen wir sie am nötigsten haben.

Daher danke ich meiner Familie aus vollem Herzen für all ihre Unterstützung, besonders während der fordernden letzten Monate. Dies gilt im höchsten Maße meiner Frau und meiner Tochter, die mich für diese Zeit so oft haben entbehren müssen und ohne deren liebevolle Fürsorge und Aufmunterungen diese Arbeit schlicht nicht möglich gewesen wäre. Dank gilt auch meinen Eltern und deren Partner, meiner Schwester und deren Familie sowie meiner Schwiegermutter für die zahllosen guten Gespräche und Aufmunterungen sowie deren fortwährendes Interesse am Fortschritt meiner Arbeit.

Ebenso danke ich meinen Kollegen, inmitten und mit Hilfe derer diese Arbeit entstanden ist. Dies gilt besonders für die enge Zusammenarbeit mit Markus Müller, aber ebenso für die anderen Mitglieder unseres kleinen, aber sehr engagierten Teams, in dem die zahllosen Diskussionen, die zu führen waren, stets zu produktiven und kreativen Ideen inspirierten.

Abschließend gilt mein Dank Herrn Prof. Dr. Brüning für die Möglichkeit, diese Arbeit anzufertigen sowie für sein stets offenes Ohr und die anregenden, fachlichen Diskussionen.

Contents

Acronyms

ACF autocorrelation function

API application programming interface

AWG American Wire Gauge

BER bit error rate

CAD computer aided design

CDF cumulative density function

CDR clock data recovery

CEI common electrical I/O

CML current mode logic

CMOS complementary metal oxide semiconductor

CPU central processing unit

CTLE continuous time linear equalizer

DAC digital to analog converter

DCD duty cycle distortion

DFE decision feedback equalizer

DFT discrete fourier transform

DPI direct programming interface

ECAD electronic computer aided design

ESD electrostatic discharge protection

ESL equivalent series inductance

ESR equivalent series resistance

EVN equivalent voltage noise

FIR finite impulse response

HPC high performance computing

IC integrated circuit

ISF impulse sensitivity function

ISI intersymbol interference

LTI linear time invariant

NIC network interface controller

NRZ non return to zero

OCM openMGT modeling framework

OCD openMGT OCM data analysis and post processing backend

OIF Optical Internetworking Forum

PAM pulse amplitude modulation

PCB printed circuit board

PDA peak distortion analysis

PDF probability density function

PDK physical design kit

PDN power distribution network

PLL phase locked loop

PRBS pseudo random bit sequence

PSD power spectral density

PSRR power supply rejection ratio

RF radio frequency

RMS root-mean-square

RNM real number model

SBR single bit response

SNR signal to noise ratio

SOC system on chip

VCO voltage controlled oscillator

VGA variable gain amplifier

VRM voltage regulator module

1 Introduction

Over the past decade there has been a strong trend towards serializer based multigigabit communication interfaces. While network interfaces between compute nodes have been using mutligigabit transmission for a long period of time already (10G Ethernet, Infiniband) and system interconnects between central processing unit (CPU) and peripheral (I/O) devices have entered the gigahertz region for quite a while, too (Hypertransport, QPI, PCI-E, S-ATA), the CPU to memory interconnects are the next domain to follow suit (HBM, HMC). This is also true for memory interfaces in the mobile processor segment[26] as well as for peripheral standards such as USB 3.0 or HDMI.

A serializer, also called *transceiver* or *SERDES*, is an electronic subsystem within an integrated circuit (IC). It consists of a transmitting and a receiving part. The transmitter acts as a multiplexer (serializer) with an output driver for the transmission channel while the receiver implements a demultiplexer (deserializer) with analog input to digital output conversion. Data is therefore modulated onto the channel or decoded from it at a higher frequency than it is accepted from or presented to the parallel side interfacing with the chip fabric.

There are two main constraints which drive the development towards ever higher data rates on the physical transmission channel between transmitter and receiver: The increase in data processing throughput of the integrated systems and the pin count limitation of chip die and package. While the first constraint is a consequence of rising internal clock rates and technology shrinks (see figure 1.1), the latter results from mechanical and thermal constraints (see figure 1.2). The continuous increase in integrated circuit performance and therefore data throughput can only be realized by providing the corresponding I/O data rate demands with modern serializer technology. An integrated system must maintain an application defined ratio between internal processing throughput and its total I/O bandwidth such that communication will produce as little of a bottleneck as possible to overall system performance. Technology shrinks not only allow to safe power and achieve more performance with the same battery lifetime but also foster a trend towards an ever growing level of integration (leading to so called *systems on chip*) whose internal processing throughput may quickly saturate the available I/O capacity. It is for this reason that multigigabit serializer technology will grow even more important in the future. This is true for all electronic systems ranging from tiny devices for the *internet of things* to mobile hand held devices all the way up to high end processors, network interface cards and the

various communication channels in high performance computing (HPC) systems.

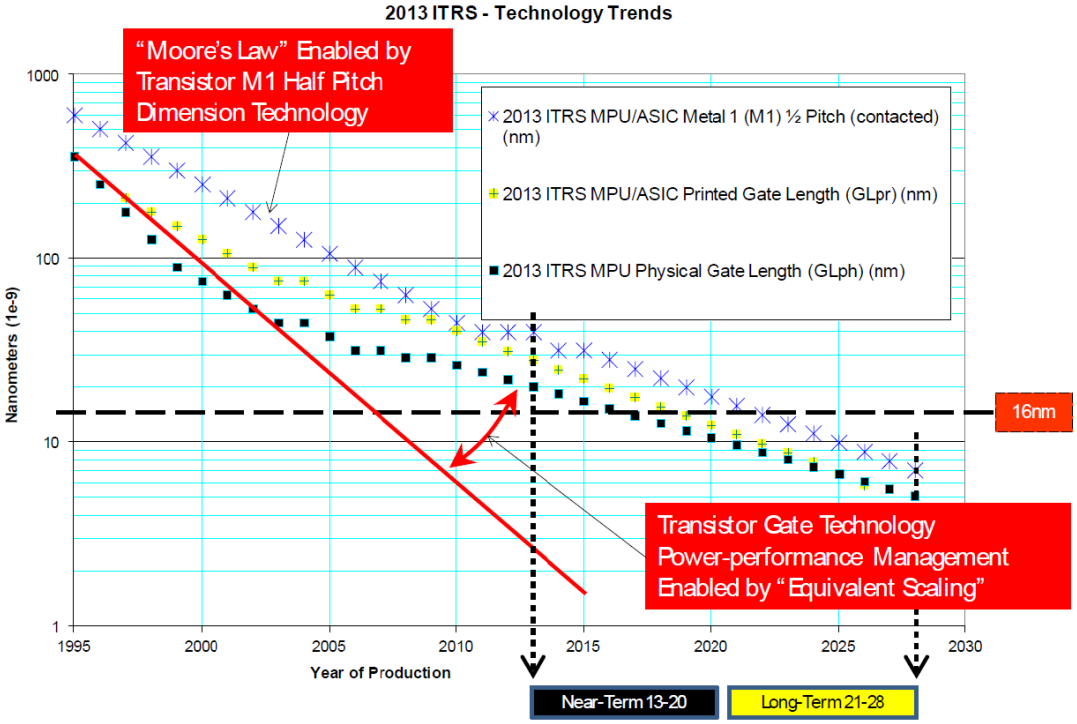

Figure 1.1: The ITRS technology projection of 2013 for feature size scaling [21]

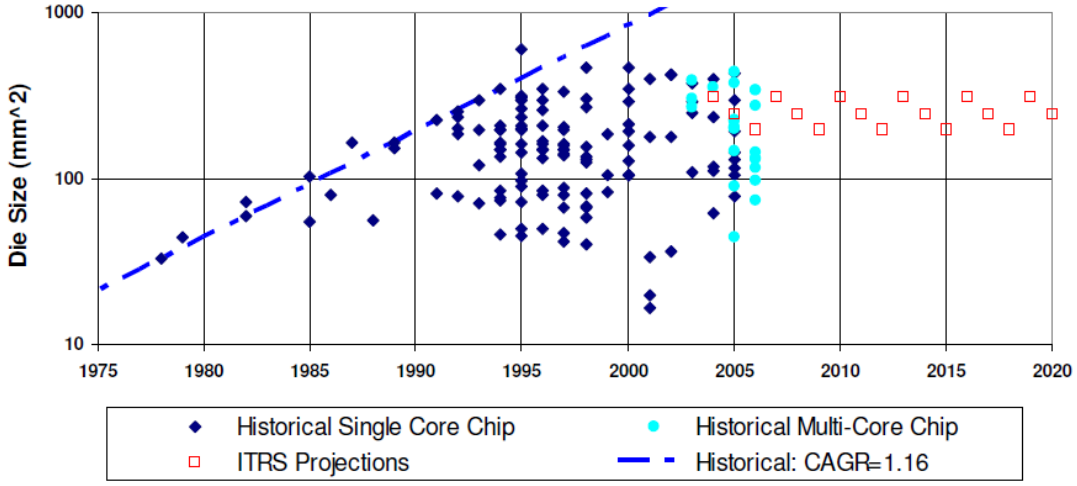

Figure 1.2: ITRS projection of total die size from 2008 [29] already revealing the fundamental thermal and mechanical constraints

As previously noted, the frequency at which data is modulated onto the transmission channel or extracted from it is related to the chip internal clock frequency via the (de)muxing ratio. This is unless more sophisticated data modulation schemes than standard non-return to zero single data rate signalling are used. Serializer circuits are usually employed to transfer information across distances far greater than that of the integrated circuit itself. Especially in high performance computing applications, there is a great variety of transmission channels. The most challenging specimen realize intercabinet or internode connections by using long copper cabling or backplane traces. Due to their high dielectric and conductor losses and the unavoidable signal degradation at every connector, all of which become worse as frequency increases, the transmitters and receivers have to compensate these effects as much as possible. Output driver and analog receiver frontend will therefore substantially grow in complexity and inevitably consume more power.

Equally challenging are chip to chip interconnects within a single node (compute system), These interconnects can either be between different CPUs, between CPUs and peripheral devices or simply from CPU to main memory. The quality of copper cables is high compared to printed circuit board (PCB) traces and its dielectric and conductor losses comparably low. Therefore, even though these node internal connections do not span as much of a distance, their malicious impact on signal quality is very significant even at shorter distances.

Another trend of past years has been to progress system integration whereever possible. This ranges from multi-die packages which may for instance combine processor and memory modules [20, 43] to chip on chip packaging with through silicon vias [65]. Despite these small distances, with the steady increase in signal frequencies to the high Gigahertz range the resulting electromagnetic wavelenghts are still comparable or below the transmission line extents. The transmission channel will therefore have to be treated and analyzed with microwave engineering approaches. Depending on operating frequency, this may be true for on-package interconnects - it certainly cannot be avoided for longer communication channels of complex topology.

Regardless of application scope, a serializer system requires a broad span of analysis and implementation tools from various disciplines. Microwave engineering techniques are required for channel and power distribution analysis alike. The output driver of the transmitter and the signal preconditioning stages of the receiver are very broadband all-analog integrated building blocks. Also, the clocking sources and clock distribution scheme are to the largest degree an analog design effort. The multiplexing and demultiplexing stages on the other hand could be approached with either analog or semicustom, digital design methods. As technologies continue to shrink, however, favouring the mostly automated semicustom digital implementation flow over manual analog design becomes increasingly appealing. Due to the smaller feature sizes of advanced submicron technology nodes, the variation in transistor properties increases and makes it difficult to meet design

specifications for analog subcomponents without additional calibration mechanisms. These mechanisms in turn rely on tunable digital to analog converters and thus digital tuning logic as do the equalization adjustment mechanisms of the all-analog stages. As a consequence, modern serializers have to be conceived as so-called *mixed signal mode* designs.

1.1 Challenges in contemporary design and analysis methodologies

One of the key challenges to the application of a serializer based communication scheme lies in the interaction of higher level protocols with the underlying, physical serializer implementation and its associated performance limits. Serializers are located at the lowest level of the communication stack, the physical layer (see figure 2.3 in section 2.2.2). Their interaction with the so-called link layer and medium access layer is very involved. Communication protocols, however, must be viewed on a larger time scale than the underlying serializer technology. While the temporal extent of a symbol on the transmission line will be in the range of a few tens of picoseconds with multigigabit signalling, the communication protocol interactions are to be analyzed in the regime of tens to hundreds of microseconds. These may include the initialization of the communication link (i.e. powering up the serializer system, synchronization of multiple serializers forming the link), the exchange of equalization presets or even link layer protocol mediated equalization adaption procedures as well as the indication of start and end of power saving or sleep modes of the link.

The design and analysis of these interactions require the availability of fast yet accurate simulation models which include information about the power state, the transistion times, frequency responses or other analog properties of individual subcomponents of the link. The challenge grows even further once tight performance or power constraints need to be met. These constraints always implicate a power and performance tradeoff at a particular point in the system. The goal is to find the right aspects where power is saved or performance increased most easily or efficiently - a process which is called *budgeting* and which only becomes possible with reasonable simulation run times and therefore very good and careful model abstraction versus performance tradeoffs.

While the modelling effort can be considered complete for those building blocks of the system which are of semicustom (all digital) nature and the respective work flows are well established in the industry, full analog elements or elements where digital and analog signals interface still have no entirely canonical work flow. For the most part, this is due to the versatility of analog components. Another reason lies in the different traditions among digital and analog electrical engineering communities. Digital designs are, to the most part, top down and text driven, while analog engineers prefer a bottom up, schematic based approach. For the specification and performance analysis of modern high speed serializers, a special level of abstraction is therefore required. On the one hand, it must

allow the definition of small, electrical subcomponents along with their required metrics, on the other hand, these subcomponents have to be embeddable into a complex system of subcomponents to form the final serializer and even further: the final link system. Within such a system, design tradeoffs, subcomponent analysis and design space exploration can be done early in the design process without the availability of more detailed models such as actual schematic implementations. This new abstraction layer should deliver hints at overconstrained (and therefore power inefficient) subsystems and should help relax specification items wherever they have a severe impact on overall efficiency or feasability. Protocol engineers are then enabled to test ideas and their impact on power savings at a very early stage and can themselves deliver valuable input to software engineers and system architects who may analyze for further repercussions on overall communication performance or efficiency. In addition to this, the tight integration requirements of modern system on chip (SOC) implementations produced and necessitate a trend towards more sophisticated hardware verification paradigms. Functional verification of digital hardware has a long lasting tradition already. The mixed mode and especially the analog sphere are catching up with this process. A serializer subsystem which may be central to almost any new high speed communication scheme can make no exception to this trend.

With the availability of suitable models for different scopes of interest, the problem of consistency between the models, the final implementation and their testbenches arises. A robust design and verification flow must ensure consistency between the various views of a component and its testbench in order to garantuee subsystem models will actually reflect the intended properties of a final implementation.

In addition to the spread in characteristic event times between the serializer subsystems and the higher level link entities, the analog properties of the serializer present further complications to system analysis. As data rates grow (and IC technologies continue to shrink), the voltage swing seen at the receiver input becomes comparable to the cumulative voltage noise magnitudes in the system itself. Also, the duration of a single bit on the transmission line becomes smaller compared to the inevitable timing uncertainties (jitter) of the clocks driving the design. Analysis of these effects require small time steps in transient simulation runs and broadband noise sources which again increases the amount of necessary computation points in classical analog simulators. Transient analysis under consideration of all noise effects will therefore quickly lead to unacceptable simulation run times and is thus unsuitable for design space exploration.

The statistical nature of noise in conjunction with the naturally uncorrelated deterministic effects make this problem even more drastic once serializer design constraints are defined such that there may only be a very small number of erroneous bits per unit time - which is typically one of the prime design targets. The probability of actually capturing a specific, malicious event is small by definition - as a consequence, the simulation time would need to be increased drastically in order to observe it. This also makes transient simulations

an ill equipped tool to analyze statistical processes in the context of transient simulations for multigigabit designs. A technique is required which ammends the standard set of operating point, transfer function, transient analysis and small signal noise analysis with a post processing environment bringing together these very different views. Additionally, some serializer subsystems require more abstract modelling views outside the classical time or frequency domain. These abstract models may also rely on information of other simulations or subsystem parameters such as the total voltage noise or timing noise and must therefore be part of the abovementioned *budgeting* and design analysis procedure.

As previously mentioned, one of the major constraints to serializer requirements is the range of physical channels that are to be supported by the given design. Transmission channels are usually described with a set of frequency dependent reflection and transmission coefficients - so called *S-Parameters*. They either result from direct laboratory measurement or can be constructed by elaborated, numerical models for a large spectrum of topologies by using modern microwave electronic computer aided design (ECAD) tools. At the same time, these microwave tools are not designed to implement large and complex mixed signal designs. There are, of course, data import and export functionalities provided by the various vendors which allows to use the most appropriate tool for each task at hand. Interoperation and data consistency between the tools then again becomes an emerging issue and mechanisms to seamlessly provide parameterizable (channel) models are, to the best of the authors knowledge, not included. From a design space exploration point of view, this forces the user to generate model files for every change in physical channel parameters - a task that can hardly be automated. Furthermore, for transient simulations, the frequency domain model needs to be converted to a suitable representation. Especially for analog, multigigabit simulations, the usual approach is to perform a Fourier transformation of the model data and deploy a continuous time convolution approach. Quite generally, this makes the channel model one of the computationally most intense components in the design with a severe impact on simulation time.

A way needs to be found which allows to leverage the ideas of analog verification and modelling of the past years [8] to speed up simulation processes and seamlessly ammend them with a powerful numerical post processing backend for advanced statistical analysis. This post processing scheme must take into account the various modelling views and subcomponent interactions. There have been many publications on these so-called *hybrid* statistical analyses of serializer systems [60, 61, 16, 45, 33, 63, 44] some of which lend themselves better to SystemVerilog and numerical backend integration than others. Additionally, they all focus on different subcomponent subsets which is why a concise overview is needed to devise an appropriate design space exploration and budgeting procedure for this demanding mixed signal mode design.

The work presented here aims at solving the challenges described thus far by using modern analog verification procedures as leveraged by the SystemVerilog [18] and Ver-

ilog/A/MS [2] language standards. Their flexibility in describing and testing mixed mode integrated circuits by using *real number* and mixed signal modelling is extended by the integration of the open source numeric software package *Octave* [12] to develop a versatile system for performance analysis, modelling, design space exploration and budgeting of mutligigabit serializer designs.

1.2 Structure of this work

This text will begin with an overview of the various subcomponents and their metrics comprising a multigigabit serializer in chapter 2. Each subcomponent and its function will be shortly introduced and contextualised with related publications. Also, the nomenclature used throughout this text, technical as well as mathematical, will be defined for later chapters and shall serve as a reference to the reader.

With the most central aspect of a serializer system being the channel, it deserves to be treated separetely in chapter 3. Its properties and the various views on signal degradation are discussed thereby highlighting the difficulties in modelling them numerically. This challenge is also expressed in the context of the so-called *bit rate capacity*, a performance metric which is still being used in HPC exascale projections. The shortcomings of this metric will be highlighted and an alternative, numerical modelling approach be presented. The resulting channel model will then be used for first, serializer implementation agnostic performance and scaling trends with respect to channel parameters. This ammends the analysis of previous work and gives valuable insight for future HPC technology projections. Also, the mechanisms of equalization in serializer systems will be introduced together with a convergence algorithm for automatic equalization adaption. In anticipation of the serializer design and analysis framework presented here, the chapter will already show some of the results to give insight into the mechanisms of equalization and estimate the time spans required by the convergence procedures.

The serializer system is built on the foundation of a framework which was jointly developed with this work and also partly published in [38]. A brief description of the framework named openMGT can be found in chapter 4. Following its introduction, it will be further extended in section 4.3 to allow for the accomodation of more compute intense system subcomponents such as the transmission channel and the receiver samplers in the context of real number based simulations. Also, a simulation performance comparison will be presented to demonstrate the benefit of the implementation presented here. In light of the growing importance of verification processes during system design, the models and testbenches developed for channel and samplers will be shown to be self-consistent.

Chapter 5 will introduce the central concepts of hybrid statistical link analysis as required to accurately verify a complete serializer system. Due to the time domain centered approaches usually taken by publications in this context, especially with regard to jitter

analysis, some of the concepts are not suitable for integration with the framework presented here. Therefore, an alternative approach is developed in section 5.3 which is based on a well established, statistical algorithm for the assessment of worst case channel properties. This algorithm is presented in section 5.2 preceding the actual openMGT/OCM budgeting procedure.

Finally, chapter 6 presents the application of the framework and budgeting procedure to an adjustable 2.5-20 Gbps serializer link architecture which was codeveloped in a team effort during the conception of this thesis.

2 Electrical serializer based multi-gigabit communication links

In the ecosystem of high performance computers, many different serializer based communication standards have been established over time. They quite often share a substantial common basis or have relaxed requirements when it comes to certain specification items. Yet, due to their very specific application ranges, some of the differences which may seem subtle at first, prohibit a particular serializer implementation to be used in another environment. This text deals with serializers that are to be used in the domain of chip-to-chip and node-to-node communication across backplanes, connectors and high quality cables. Some of the results may be useful for or extendable to other contexts such as on-chip communication or electro-optical systems as well. The framework developed throughout the next chapters will be designed with extensibility in mind. However, the system overview and nomenclature as well as the background on electrical transmission lines and equalization presented here puts its emphasis on the first mentioned use case.

2.1 Mathematical definitions and relations

This section presents a clarification of the mathematical conventions and the nomenclature used throughout this text. Especially the radio frequency (RF) and microwave community tend to use very different notations which is why an attempt is being made to unify the approaches as much as possible. Whenever formulas from papers are used and cited, their notation will be given in the form presented here.

- The complex conjugate to a variable $a \in \mathbb{C}$ is a^*
- Vectors $\boldsymbol{v}$ are written boldface and are lowercase
- Matrices $\overline{\boldsymbol{M}}$ are written boldface with a bar over them and are uppercase
- Metrics and parameters of the OCM link budgeting procedure $\boldsymbol{b}_{\mathbf{item}}{}^{\ddagger}$ are written boldface italic throughout this text irrespective of their actual dimensionality or nature. This serves as a reference for Appendix A where the parameters used for the

link budgeting procedure developed throughout this text and specifically in chapter 5 are again listed as an overview.

- A function f which is continuous with respect to its argument x is written as $f(x)$
- A function f which is discrete with respect to its argument x is compactly written as vector $\boldsymbol{f}$ while specific values of the function are denoted by f_x
- A Fourier transform pair is short handedly written as $f(x) \circ\!\!-\!\!\bullet\, F(p)$.
- The Fourier transformation of a function $\cdot$ is denoted by $\mathscr{F}\{\cdot\}$ and the inverse Fourier transformation by $\mathscr{F}^{-1}\{\cdot\}$
- The convolution $h(x)$ of two functions $f(x)$ and $g(x)$ is described and defined as

$$h(x) = f(x) * g(x) \equiv \int_{-\infty}^{\infty} f(x)^* g(x - x')dx' \tag{2.1}$$

If f is a continuously valued function of a continuous variable x the transformation set is given by

$$\begin{aligned} \mathscr{F}\{f(x)\} &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x)e^{-jpx}dx &= F(p) \\ \mathscr{F}^{-1}\{F(p)\} &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(p)e^{jpx}dp &= f(x) \end{aligned}$$

where the continuous function F will then of course depend on a continuous argument p. In the context of real number modeling and numerical statistical analysis, however, all function arguments will necessarily be of discrete nature. In this case the function f and F are represented by vectors $\boldsymbol{f} = (f_0, ...f_{N-1}) \in \mathbb{C}^N$ and $\boldsymbol{F} = (F_0, ...F_{N-1}) \in \mathbb{C}^N$ where $f_n = f(x_n)$ and $F_n = F(j\omega_n)$.
The discrete fourier transform (DFT) and its inverse can be calculated according to

$$\begin{aligned} F_n &= \sum_{k=0}^{N-1} f_k \cdot e^{-\frac{2\pi j \cdot kn}{N}} \\ f_n &= \sum_{k=0}^{N-1} F_k \cdot e^{\frac{2\pi j \cdot kn}{N}} \end{aligned}$$

respectively. Wherever it is beneficial to a more comprehensible discussion, the text will

use the continuous description and deviate from this path only to highlight important aspects of a particular implementation. The two vectors $\boldsymbol{f}$ and $\boldsymbol{F}$ are associated with their argument vectors $\boldsymbol{x}$ and $\boldsymbol{\omega}$. The following relations hold for every function argument vector pair ($\boldsymbol{x}$ / $\boldsymbol{\omega}$):

$$x_0 = dx = (x_{n+1} - x_n) = \frac{2\pi}{\omega_{N-1}} \qquad \text{and} \qquad x_{N-1} = \frac{2\pi}{\omega_0} \tag{2.2}$$

and conversely

$$\omega_0 = d\omega = (\omega_{n+1} - \omega_n) = \frac{2\pi}{x_{N-1}} \qquad \text{and} \qquad \omega_{N-1} = \frac{2\pi}{x_0} \tag{2.3}$$

Furthermore, the following functions are defined:

- The power spectral density (PSD) of a function $f(x)$ is denoted by $S_{ff}(j\omega)$ and equates to

  $$S_{ff}(j\omega) = \mathscr{F}\left\{|f(x)|^2\right\}$$

- The autocorrelation function (ACF) of a function $f(x)$ is denoted by $r_{ff}(x)$ and is defined as

  $$r_{ff}(x) = \lim_{X\to\infty} \frac{1}{2X} \int_{-X}^{X} f(x)^* f(x - x')dx' \tag{2.4}$$

  $$r_n = \lim_{N\to\infty} \sum_{k=1}^{N-1} f_k^* f_{k+n} \tag{2.5}$$

  in the continuous and discrete case. The ACF is a measure for the resemblance of a function with itself.

- In this text the Gaussian probability density function (PDF) is used in the following form:

  $$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{2.6}$$

  with the standard deviation σ and the mean value μ.

Whenever there is a parameter σ_x it signifies that the underlying values of the statistical function x exhibit a Gaussian distribution whose standard deviation (or equivalently the root-mean-square (RMS) value of x) is given by its value. The error function and

complementary error function are defined as

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{-\infty}^{x} e^{-t^2} dt$$
$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x)$$

which are the integrals of a Gaussian distribution with unity standard deviation up to and starting from a given point x respectively and thus represent cumulative probabilities such as required when defining the bit error rate (see below).

Oftentimes, conversions of power spectral densities from frequency to phase space will be required. A conversion between phase and frequency domain can be made with respect to a center frequency f_0 due to

$$\Delta\varepsilon(t) = \varepsilon(t) - f_0 = \frac{1}{2\pi} \frac{d\phi(t)}{dt}$$

and therefore

$$S_{\varepsilon\varepsilon}(f) = \frac{1}{f_0^2} S_{\Delta\varepsilon\Delta\varepsilon} = \frac{f^2}{f_0^2} S_{\phi\phi}(f)$$

Phase noise is usually given as a single sided spectral function (meaning from 0 to ∞). For oscillators the definition of the phase noise power spectral density as $\mathscr{L}(f) = \frac{S_{\phi\phi}(f)}{2}$ is also common. In figures, its magnitude is always given relative to the power at the central carrier ($\frac{\mathrm{dbC}}{\mathrm{Hz}}$). IEEE calls $S_{\phi\phi}(f)$ the phase instability and $\mathscr{L}(f)$ the phase noise. We will make no semantic distinction here.

2.2 Link system overview and nomenclature

Figure 2.1 presents a condensed block diagram of the communication system as it is being investigated throughout this text. This section will give an overview of the subcomponents, their function and their most central performance metrics. In the context of link budgeting, this will provide the orientation needed when combining the various subcomponent interactions to derive a final metric for overall system performance. This final metric is called the bit error rate (BER) of the communication system. It comprises all deterministic and random influences on information propagation from transmitting to receiving side. As such, it is a statistical quantity and defines the probability of detecting an erroneous bit at the receiver output. It also marks the starting point of the subcomponent and metric discussion in this section.

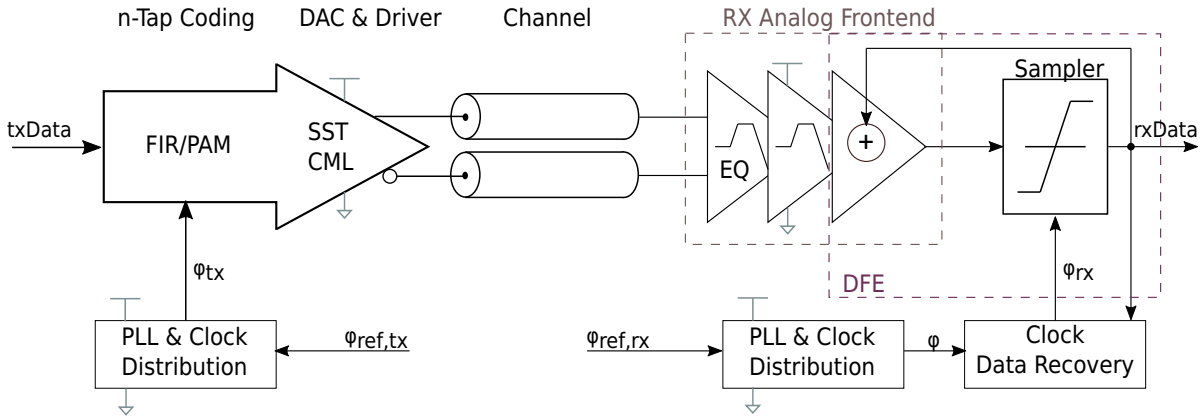

Figure 2.1: The system components of a multi-gigabit serial link as they will be used, described and modeled throughout this text

2.2.1 The bit error rate

Figure 2.2 shows an idealistic and *deterministic* eye diagram as it may result from overlaying a few of the worst case patterns (sequences of logic ones and zeros) seen at the analog input of the receiver samplers after passing through the system. We will call the path from transmitter (TX) output to receiver (RX) sampler input the *system channel*. It comprises the actual physical transmission medium and all signal equalization stages. The eye diagram is assumed to have been obtained either by a noiseless transient simulation or by statistical methods (see section 5.2). In this particular example, the line coding (see subsection 2.2.2) is chosen to be non return to zero (NRZ) for simplicity. However, the arguments below remain valid even for more complex coding schemes:

All deterministic effects are bounded which makes it possible to generally calculate the probability of seeing a particular voltage level at a specific point in time. Quite actually, the single colored eye diagram of figure 2.2 is a three dimensional map with the third dimension specifying the probability of seeing the given voltage at the specific time instant. Oscilloscopes, papers and statistical channel analysis in this text therefore use a color scheme to highlight this aspect of the eye diagram which is left out in the figure above only for simplicity. Without random noise sources, the deterministic worst case eye of the channel equalized by the transmitter and receiver filters and impeded by residual offsets and duty cycle distortion would be the final metric. The goal would then be to design a system which exhibits an eye diagram of smallest height and width for the required transmission channels (assuming a perfect receiver sampler, see section 4.5). In such a case, the efforts of equalization and therefore power consumption would be kept at a minimum while it was garantueed to never receive a single bit in error.

The unavoidable presence of random noise sources in the system, however, makes

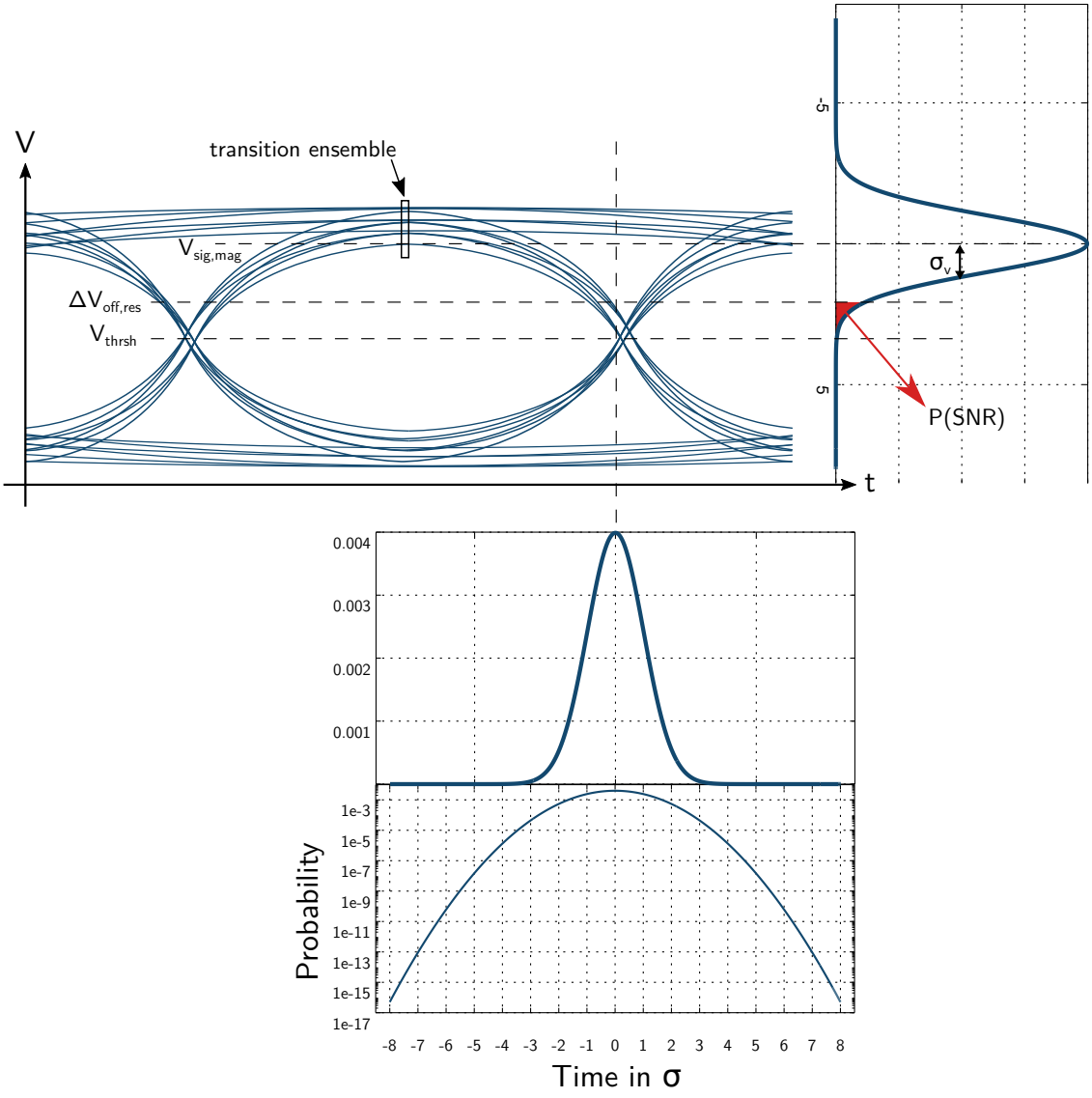

Figure 2.2: An idealistic, reduced representation of a *deterministic* eye diagram as captured at a sampler input along with the Gaussian distributions of voltage and timing noise in linear and logarithmic scale normalized to their standard deviation.

analysis much more complicated. Random noise sources are unbound and in the context of this text are all based on underlying physical processes that are wide-sense stationary and ergodic. The first restriction ensures consistent process behavior independent of the initial value and point in time of the noise source state. The latter garantuees that the process will eventually assume all possible internal states which ensures that by observing the process over time, its *PDF* can be derived. The combination of all unbound statistical processes within the system, such as thermal noise, shot noise or flicker noise (which do themselves not adhere to a Gaussian distribution except for thermal noise) will eventually result in a Gaussian distribution of timing and voltage noise due to the central limit theorem of statistics. System internal and external noise sources affect the sampler input signal voltage seen at a particular point in time. Depending on the view, this can either be considered a voltage noise or a timing noise (jitter). The slew rate of the signal translates one into the other and it depends on the subsystem being analyzed which of the views is more convenient. The presence and interaction of both types of perturbation to the eye diagram is indicated by the two Gaussian probability density functions in figure 2.2. In addition to the linear scale of the PDF which is normalized to the standard deviation σ of the process, its logarithmic representation is also shown. It highlights the unbound nature of the process and shows that extreme noise contributions both in voltage and time dimension are possible albeit at very low probabilities. For better visibility, a large σ compared to deterministic eye width and height was chosen here.

For a noiseless system, the performance metric is straight forward to define, the eye width at the decision threshold (usually chosen to be at 0) and its height for a specific instant in time can be derived with ease. The system would be said to achieve the given eye opening at a randomly low error level. With unbound noise sources on the other hand, the eye diagram is closed by definition. It is therefore necessary to define beforehand which probability of receiving a bit in error is acceptable - the so called BER. An on-die transmission system between higher level cache and processor instruction prefetch stage for instance has only little signal distortion and comparably high transmission levels. It is therefore rather easy to achieve BER levels much lower than 10^{-30} which makes it unlikely to even receive a bit in error during the lifetime of the processor itself.

As transmission line properties decrease signal voltage levels to magnitudes comparable to those of the noise sources in the system (thus decreasing the so-called signal to noise ratio (SNR)), more realistic target BERs need to be chosen. In fact, knowing the achievable SNR at the receiver sampler (which is also directly related to the power consumption of the transmission system), directly determines the achievable BER level. It is the particular serializer application along with the constraints at higher OSI layers (see section 2.2.2) that call for a specific lower BER bound. For instance a serial link with a transmission rate of 10 Gb/s shall on average only produce a single maldetected bit per minute such that the network protocol retransmission procedure does not have a severe impact on total link

performance. This is equivalent to a bit error rate of approximately $1.7 \cdot 10^{-12}$.

From figure 2.2 it can be seen that it is the integral of the portion of the Gaussian process reaching below the decision threshold which gives the cumulative probability of actually receiving a bit in error. The connection between SNR and BER can therefore be derived to

$$BER = P(SNR) = \frac{1}{\sigma_V \sqrt{2\pi}} \int_{-\infty}^{SNR \cdot \sigma_V} e^{-\frac{1}{2}\left(\frac{x - V_{sig,mag}}{\sigma_V}\right)^2} dx = 0.5 \operatorname{erfc}\left(\frac{SNR}{\sqrt{2}}\right) \tag{2.7}$$

The signal to noise ratio in figure 2.2 can pessimistically be given by

$$SNR = \frac{V_{sig,mag} - V_{thrsh} + \Delta V_{off}}{\sigma_V}$$

In the example of the 10 Gb/s link above, an SNR of about 7 would need to be achieved at the sampler. This is pessimistic in the sense that here we assumed convolution of the Gaussian distribution with the worst case transition alone. However, this convolution would need to be performed with the ensemble of all transitions passing through the sampling time instant t_s in the figure (highlighted by the black box) which would of course reduce the relative impact of the worst case transition in accordance to its probability of actually occuring in a given bit pattern (see chapter 5 for more information on this process). Note also, that the probabilites of the Gaussian plots in figure 2.2 are neither scaled with the probability of the worst case transition nor can the probability of receiving a bit in error directly be read from the graphs. This would additionally involve the abovementioned integration over the ensemble of all noise source magnitudes which could potentially push the voltage level below the decision threshold.

While it is straight forward to analyze the eye diagram in terms of voltage noise (and therefore uncertainty in voltage amplitude, hence voltage error), uncertainty with respect to instants in time (timing errors) need to be defined and treated more carefully:

There are mainly three different descriptions of timing error which are all interrelated. They are the so-called *phase jitter*, the *period jitter* and the *cycle to cycle jitter*. The latter two are important when analyzing autonomously oscillating systems or parallel bus systems where a known timing between serveral lanes is important due to common retiming on the receiving side. With serial links as discussed here, each lane is considered a separate system. Retiming can and will be delegated to other layers of the communication stack such as the medium access layer (see section 2.2.2). The serial link transmits symbols in well defined time intervals called bit period T or sometimes unit interval UI. The existence of T is an idealized assumption and helps to quantify the time interval error (TIE) which is due to residual equalization errors, residual offset errors or phase noise of the clock sources driving the gating structures responsible for creating the potential differences on the transmission line that ultimately represent the information (bits). It is in this sense that

we can define the phase jitter $\phi_n = t_n - nT$ where $n \in \mathbb{N}$. While nT would be the moment in time where the ideal signal crossing would occur if the information modulated onto the transmission line were actually changing, t_n is the actual point in time observed for the transition through the reference level. From a known phase noise PSD $S_{\phi\phi}(\omega)$ or its dual autocorrelation function, the RMS time interval error (jitter) can be derived to

$$\sigma_\phi^2 = \frac{4}{\omega_0^2} \int_0^\infty S_{\phi\phi}(\omega) d\omega = \frac{2}{\omega_0^2} R_{\phi\phi}(0) \tag{2.8}$$

where $\omega_0 = \frac{2\pi}{T}$.

The SNR is a voltage domain quantity. For jitter, however, the same derivation can be made with respect to the time domain. This is indicated by the Gaussian distribution below the eye diagram. The decision threshold in this case is the (ideal) sampling instant usually located at the center of the eye. The Gaussian tails to be integrated give the probability of sampling a bit pre- or succeeding the actual bit to be sampled. It is for this reason, that there must always be a triplet of information given to describe the performance of the overall system: the resulting eye width $\boldsymbol{e_w}^\ddagger$, the eye height $\boldsymbol{e_h}^\ddagger$ and the $\boldsymbol{BER}^\ddagger$ level at which the prior two values were obtained. As the processor example above indicates, a BER for a serial tranmission system should always be given with respect to channel attenuation at the Nyquist frequency (and hence serializer data rate) and the system testing pattern used (a PRBS sequence for instance) in cases where a non-statistical analysis is made. Oftentimes, even more information like channel reflections are given too, to highlight the importance and effectiveness of more involved equalization schemes such as the DFE (see section 3.3).

2.2.2 Interaction with higher communication layers

Figure 2.3 depicts the lowest three layers of the *open systems interconnection (OSI) model* which are called *physical layer*, *data link layer* and the *network layer* respectively and define the constraints for the serializer system. The bulk of digital hardware components which need to implement certain aspects of these layers are omitted in figure 2.1. These include the muxing and demuxing structure to adapt to on-chip data width and rate, the buffering structures, power down, idle modes as well as offset and equalization calibration logic. They are all not shown as they are handled by the well established digital description and implementation work flow which is ammended by the real number modelling framework (see section 4.2 of chapter 4 for details). Also, these aspects, albeit integral part of the serializer itself, can conceptually be attributed to the higher layers as well. The specification items listed in the diagram focus on those aspects of the higher two layers which conversely exhibit an interaction with the serializer performance at the lowest level. The

Network Layer	Packet definitions	Test patterns, NOP/Skip, Data/Control
	Packet error correction	Retransmission, Buffering
Link (MAC) Layer	Data coding	XB/YB, Scrambling, DBI
	Link maintenance	Presence detection, calibration phase, equalization, data rate negotiation, power down and sleep modes, recovery
	Error correction	CRC, FEC, fault tolerance
Physical layer	Line coding & Signal levels	NRZ (PAM-2), Duobinary, PAM-4
	Impedance	40/80 to 60/120 Ohms
	Channel type	Single-ended, Differential, Termination, Coupling (AC/DC)

Figure 2.3: A selection of specification items and the OSI layers they are associated with. The items of the higher two layers in the diagram show interaction with the lowest layer. Higher layer analysis and design space exploration therefore requires an understanding of their influence on serializer performance.

most substantial interdependences are of course located at the data link layer which defines the low level link protocol but even at the network layer, there are design decisions to be made that have a direct impact on serializer implementation and performance evaluation.

The design choices at the physical layer do not all directly mirror in the rather abstract view of figure 2.1. Most importantly, the *channel types* to be supported need to be well specified. This quite generally includes whether data is transmitted electrically or optically, whether transmission is wireline based or over-the-air and whether a bandlimited modulation technique or broadband communication is being used. In the context of wireline, broadband electrical communication as discussed here, one of the most central aspects is the *impedance domain* in which communication occurs (see also chapter 3). While for on-chip communication, this domain may be rather high impedance, long transmission channels across backplane and cabling generally use industry standard values in the range of 40 to 80 Ω. The impedance domain is also very important with respect to the coupling and termination scheme. While signal coupling can electrically either be done directly, capacitively or inductively, the termination schemes have a far greater variety (as evident from the vast number of different signaling schemes supported by modern FPGA IO cells for instance). For on chip communication, single ended capacitively coupled schemes without explicit termination have become popular [41] while for long-haul backplane communication as discussed here, differential signaling with termination at both ends of the channel is widely used. This is primarily due to the superior noise immunity and independence of any reference level between transmitter and receiver resulting from this scheme. AC coupling is primarily used once the transmission channel crosses connectors or

even systems (compute nodes for instance) while DC connections are preferred whenever tight control of the entire system to be implemented is possible. An example for this would be a memory module (such as HMC) connected to a CPU on the same board or even package where the power as well as the reference clock distribution are part of the systems design space.

In addition to impedance the *line coding and signal levels* for data transmission and reception need to be defined. The line coding is specified in the physical layer and one of the earliest design decisions of all. In this text, the focus solely lies on NRZ coding. This is, for the most part, due to its wide popularity and the interoperability requirements of the serializer whose development was assisted by this work. Another more complex choice for line coding may have been a duobinary or ternary coding scheme. Essentially, these codes exhibit a vastly different power spectral density (see section 3.4) with a redistribution of signal alphabet power in favor of lower frequency. Especially in high loss, long haul channels, this can substantially facilitate equalization efforts or the feasibility of power constraints. A much more involved choice offers a scheme such as a PAM-N code, where PAM stands for pulse amplitude modulation (PAM). A transceiver which utilizes NRZ line coding is also said to be using a PAM-2 scheme. There have been publications on PAM-4 systems [4, 32, 28, 13] which, with the advent of 100G Ethernet finally arrive at a commercial level as well. There is a substantial increase in design complexity associated with PAM-4, especially at the receiver (refer to caption of figure 2.12). Not only does the number of samplers and clocking resources increase. Also, equalization schemes and clock data recovery analysis are much more involved. The benefit ultimately lies in the relaxed equalization requirements compared to a PAM-2 system with the same baud rate. The baud rate is the number of symbols transmitted per second. An NRZ code encodes a single bit per bittime with its two voltage levels. A PAM-4 code on the other hand transmits two symbols per bit time with its four defined voltage levels.

At the link layer, one of the most influential decisions with respect to serializer performance and constraining is the *data coding*. Popular coding schemes are XB/YB schemes in which words of data to be transmitted are translated from X to Y bits with Y being greater than X. Concepts of how this transformation is carried out differ substantially for the various choices of X. In the past, one of the most popular choices has been 8B/10B coding [64] which used a predefined translation table under omission of some of the 10 Bit codes to guarantee a DC balance in the signal over a well defined run length. The DC balance constraint was meant to avoid a shift of the common mode voltage at the receiving samplers which would result in a reduction of the effective signal to noise ratio. At the same time, there is a guaranteed number of transitions per unit time so that AC coupled transmission lines may be used. Also, the clock recovery mechanisms all rely on tracking signal transitions and will therefore leave the ideal sampling instant (i.e. loose lock) when a specific lower bound for the transition density can not be upheld. When designing the

clock recovery circuit of a receiver, this is a very central aspect which requires thorough analysis. 8B/10B coding with its high transition density is a rather conservative choice in this regard and comes at a pretty substantial cost: only 80 percent of all bits transmitted actually convey user information. This is why in recent years, other combinations of X and Y have become more popular. The two most prominent are 64B/66B (10 Gigabit Ethernet / Fibre Channel) as well as 128B/130B (PCI-E 3.0). In both cases, there is no explicit transformation table. To ensure a statistical DC balance, the transmitter *scrambles* the data with a linear feedback shift register of a predefined polynomial while the receiving side inverts this process to recapture the original data. Theoretically, it is possible to force a scrambler into producing an indefinitely long sequence of static bits with a well chosen succession of input data. Usually, user data exhibits a good level of variability which the scrambling in turn will convert to a well randomized output bit stream (and therefore to a spectrally white frequency distribution). This may also be important for equalization training algorithms, especially the widely popular SS-LMS algorithm (see section 3.3) which rely on an equal symbol probability distribution.

Link initialization and management are also vital aspects of the link layer. In this context, the procedures in which a link is powered up or down, calibrated (to compensate circuit mismatches) or even set up with respect to its data rate and equalization need to be defined. If automatic adaption schemes are to be implemented in hardware, one of the key aspects to investigate is how to perform digital loop based equalization procedures and how these calibration loops respond to an elevated, initial bit error level. Another important aspect in this context is the definition of *link operability*. Usually, a link is said to be operable once it reaches the predefined bit error rate (see next subsection). The bit error rate cannot achieve low levels of statistical insignificance in all transmission domains, especially not within the domain primarily described in this text. The link layer therefore also needs to implement an error detection and recovery mechanism if the overall communication stack specification aims to avoid severe latency penalties as would be incurred if retransmissions of faulty data were delegated to higher OSI layers. Error detection and correction schemes can for instance be realized by forward error correction (FEC) or by cyclic redundancy checks (CRC). Fault recovery when failing to correct the received data may include the retransmission of the faulty packet. A thorough analysis of which error levels still allow to maintain an operable link within given performance specifications may allow to tweak overall power consumption and aid in the development of robust link transmission protocols. This again requires a serializer model which is closely tied to its physical implementation but simulates magnitudes faster.

The *network layer* may also benefit from a concise transceiver model. When defining test patterns and the overall package structure of the network, a thorough analysis may help to increase the overall power efficiency of the system and may expose new ways of implementing network power states and utilization awareness. This in turn helps to avoid

overconstraining the metrics for the serializer itself.

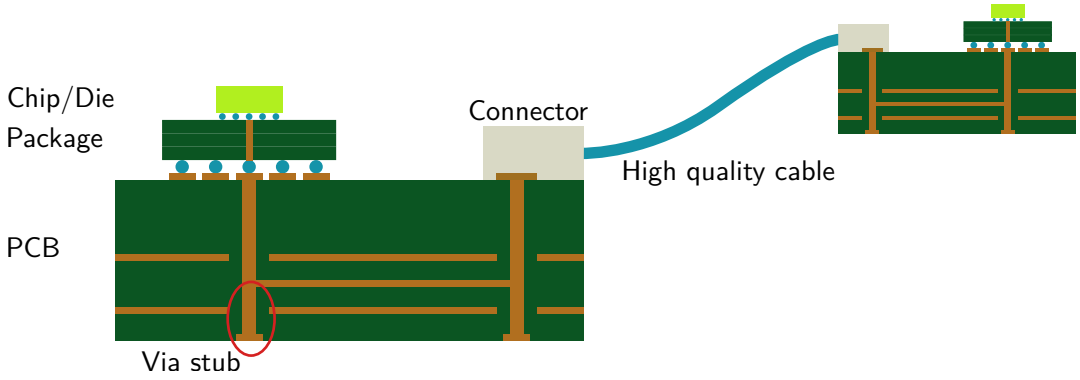

Figure 2.4: Physical system topology

2.2.3 Power distribution

Figure 2.1 hints at the various power distribution networks (PDNs) of the subsystems. The PDN may have a strong impact on system performance, especially in highly integrated SOC environments. Whenever the small-signal properties of a circuit are of importance or in cases where the signal to noise ratio is of concern the influence of the power supply cannot be safely ignored. This usually excludes digital logic as long as excessive switching noise or supply voltage drops due to a lack of proper decoupling or series resistance (IR) analysis does not pose a problem.

Figure 2.5 shows a typical impedance versus frequency function and the domains which dominate the response in the various regions of the plot as it would be expected from a mechanical arrangement such as the one shown in figure 2.4. The example is taken from an analog supply voltage rail of a custom built hybrid memory cube (HMC) test board. The electromagnetic extraction of the PCB board was performed in conjunction with an ECAD vendor supplied capacitor model database while the chip vendor supplied the combined die and package input impedance.

At lowest frequencies, the output impedance of the voltage regulator module (VRM) sets the lower bound of the power distrubtion network impedance Z_{PDN}. The output of the VRM is decoupled with very large capacitances (usually electrolyte or tantalum capacitors) which typically have a fair amount of equivalent series inductance (ESL) due to their mechanical size but are required to have very low equivalent series resistance (ESR) in order to maintain the good output impedance characteristics of Z_{PDN} at the lowest frequencies. The elevated level of ESL, of course, quickly forces their impedance to grow as frequencies increase. Therefore, small size ceramic capacitors on the PCB and in close

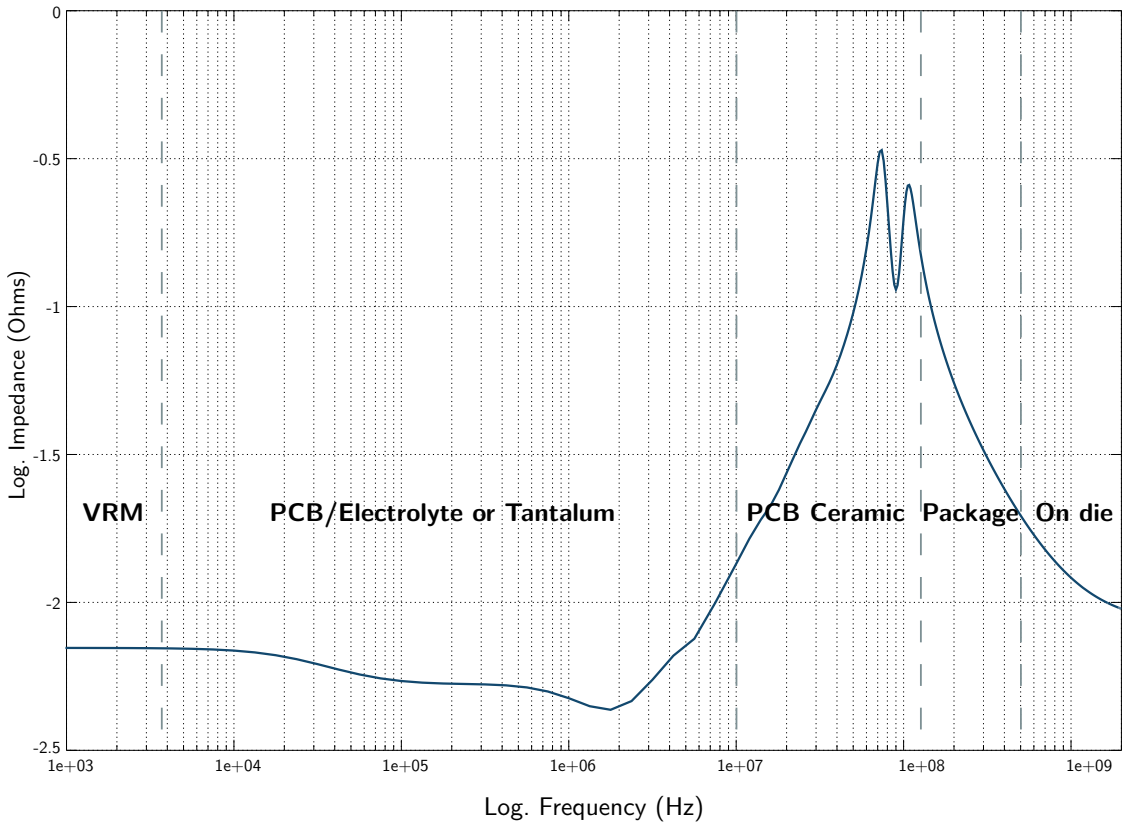

Figure 2.5: A typical power distribution network impedance characteristic

proximity to the current sinks are used to decrease Z_{PDN} at higher frequencies. If the number of capacitors on the PCB are to be kept small, there is usually an impedance peak in the region around (low) hundreds of MHz. This peak arises from the interaction of chip package inductance and PCB capacitance. Low ESL capacitances of small footprint and in close proximity to the chip package can be used to dampen the peak. Also, on-package capacitances may be used to attempt the same. However, oftentimes onboard capacitances suffer from additional ESL due to vias, especially on thick multilayered boards while on-package capacitances are rather small due to limitations to their mechanical size. Therefore, on-board capacitances are often effective in the tens of MHz region, while on-package capacitances take an effect in the region of hundreds of MHz and slightly beyond. The impedance at very high frequencies is dominated by the actual chip (die) power distribution and decoupling network. On-die capacitances have virtually no equivalent inductance but may suffer from increased series resistance if not carefully tied to the supply network with large metal strips and sizable via count. It is the averaged ESR of the chip which dominates the high frequency behavior of Z_{PDN}. It has become custom practice to separate the power supplies of the PLL from the rest of the system. Transmitter and receiver are

usually tied together in a common domain. In order to minimize interactions between the digital, complementary metal oxide semiconductor (CMOS) and therefore switching noise dominated part and the analog, often current mode logic (CML) dominated and noise sensitive part of the design, a separation of digital and analog supplies on the level of chip and package has also been used. Additionally, with a shrink in technology feature sizes, the core voltage decreases, too, due to smaller gate oxide thicknesses. While 65 nm nodes can still be operated with up to 1.2 Volts, the 22 nm node forces designers to work with supplies as low as 0.7 Volts. This quickly becomes a problem for all I/O standards and specifically to serializers (see also subsection 2.2.5). The alternative of using thick-oxide transistors when designing for smaller technology nodes neccessitates a further supply and PDN which leads to design challenges especially on the packaging level but may have a favourable impact on the supply noise characteristics in conjuction with other specific design choices (see again subsection 2.2.5). As can be seen from this discussion, there is a lot of information required to obtain a meaningful Z_{PDN}. In addition to the vendor supplied characteristics of capacitors and VRM, at least a 2.5D field solver based extraction of the power distribtion network (PCB and package) is required. Also, a good estimate of the total die capacitance and ESR is needed as well. Typically this kind of information is only available very late in the design phase. Therefore, it is common practice to model the PDN as a bandwidth limited thermal noise source with a power spectral density of either

$$S_{\mathrm{VV,\,LP}}(\omega) = \frac{\sqrt{\pi}(1+\sqrt{2})\sigma^2_{\mathrm{vn,pdn}}}{\omega_{\mathrm{3db,pdn}}} \cdot \frac{1}{1+\left(\frac{\omega}{\omega_{\mathrm{3db,pdn}}}\right)^2} \tag{2.9}$$

or (much more unphysical and even discontinuous)

$$S_{\mathrm{VV,\,Box}}(\omega) = \begin{cases} \frac{\sigma^2_{\mathrm{vn,pdn}}}{\omega_{\mathrm{3db,pdn}}} & , 0 \le \omega < \omega_{\mathrm{3db,pdn}} \\ 0 & , \text{else} \end{cases}$$

where in both cases $\boldsymbol{\sigma_{\mathrm{vn,pdn}}}^{\ddagger}$ is the actual in-band RMS voltage noise and $\boldsymbol{\omega_{\mathrm{3db,pdn}}}^{\ddagger}$ the PDN bandwidth. The additional factor for the lowpass filter type is chosen such that in both cases we obtain

$$\int_0^{\omega_{\mathrm{3db,pdn}}} S_{\mathrm{VV}}(\omega)d\omega = \sigma^2_{\mathrm{pdn}}$$

and thereby ensure that the noise power contribution made by the PDN to the system is well defined by the parameter $\sigma_{\mathrm{vn,pdn}}$. Note that here, we drop the notion of the PDN having an impedance alltogether and think of it plainly as a source of noise power. This noise power would normally originate from the interaction of the systems current consumption $S_{\mathrm{I}}(\omega)$

(given in its spectral form) with the PDN impedance which is also a complex quantitiy. The voltage spectral density seen at the power supply would then be given by

$$S_{\mathrm{V}}(\omega) = S_{\mathrm{I}}(\omega) \cdot Z_{\mathrm{PDN}}(\omega)$$

and its equivalent voltage noise can be determined from the resulting PSD as $S_{\mathrm{VV}}(\omega) = |S_{\mathrm{V}}(\omega)|^2$. As mentioned above, a good estimate of both $\boldsymbol{S_{\mathrm{I}}(\omega)}^{\ddagger}$ and $\boldsymbol{Z_{\mathrm{PDN}}(\omega)}^{\ddagger}$ can generally be obtained with substantial effort and can potentially be included in budgeting procedures. Conversely, it is possible to use budgeting approaches to produce meaningful constraints to $Z_{\mathrm{PDN}}(\omega)$ as board and package implementation design input. Care must be taken, however, to not delegate too much of the noise reduction efforts to system designers as their design space is much more limited compared to the IC design itself. The equivalent noise source approach of course assumes that the noise seen at the power supply is completely uncorrelated with serializer events. Evidently, this is especially untrue for CMOS type logic of which an increasing amount will be found in the serializers of the nodes and years to come. While for the initial system design phase, the simple model serves its purpose, the final design verification and sign-off should rely on more elaborate methods. An intermediate solution to the problem is the definition of an impedance mask and the deduction of approximate time resolved current consumption as part of the modeling process.

2.2.4 Phase locked loop

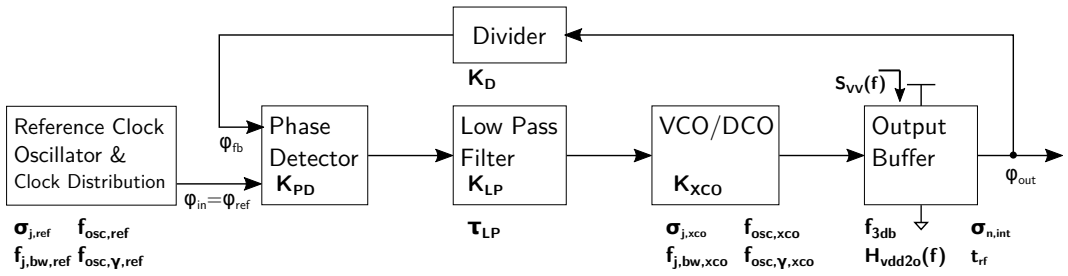

Figure 2.6: A simplified block diagram of a phase locked loop (after [34]) with the subcomponent metrics as used for link budgeting in this text

As depicted in figure 2.4, a transmitter and receiver pair which form the active part of a communication channel are surely located on different dies. In some applications such as in system area networks (PCI-e, QPI) and backplane interconnects, their PLLs may share a common reference clock ϕ_{ref}. However, even in these cases, the clock distribution channels from the common reference oscillator to the respective dies usually are not identical. The reference clock channel jitter amplification properties may therefore differ, too (see

section 5.3.4) which may in turn lead to differing characteristics for the two PLLs. In HPC networking applications, the reference oscillators are certainly not identical whereas the clock distribution channel usually is (if a reference oscillator is located on the network interface controller (NIC) PCB). In this case, a single PLL model for both sides is sufficient. The reference oscillator together with its clock distribution channel is not shown in figure 2.1 as it is a part of the phase locked loop (PLL) model here. The task of a PLL is to generate a high frequency, low jitter reference clock for the transmitter and receiver clock dividers, its (de-)mux stages, the transmitter output driver and the receiver samplers. It is distributed to transmitter and receiver via the on-chip clock tree. The effect of on-die clock distribution is logically assigned to transmitter and receiver domains and will be discussed below. The effect of clock distribution from the high precision reference oscillator (usually a quartz with an oscillation frequency of a few MHz) can potentially be integrated with its model representation directly. Figure 2.6 shows a typical PLL block diagram together with the metric symbols used for budgeting. The design space of these subcomponents is quite large and well beyond the scope of this text. Fortunately, irrespective of the exact implementation, it is usually possible to model a PLL as a second order transfer function in phase space. Ultimately, for the system presented here, the relevant high level metric of the PLL is the phase noise power spectral density $\boldsymbol{S_{\phi\phi}(\omega)}^{\ddagger}$ *including* the effects of the PLL power supply. If it is known beforehand and the PLL is no part of the design budgeting process, it is the single piece of information required for deriving constraints for other subcomponents. If the PLL design effort shall be part of the constraining efforts, however, a good or at least rough approximate shape of the phase noise PSD is of importance. It is the spectral content of this jitter that will impede the transmitter output, be potentially amplified by the channel and be finally processed by the clock data recovery circuit of the receiver. Therefore, it will have a significant impact on the system performance (see section 2.2.6). In a final implementation $S_{\phi\phi}(\omega)$ will be obtained by a periodic steady state and periodic noise simulation as offered by modern RF Spice simulators. The sensitivity of the PLL to noise on the power supply can also be obtained in this way - the necessary analysis is called the periodic transfer function (PXF) simulation and is also beyond the scope of this text. As is the case with detailed PDN metrics, these simulations can only be obtained fairly late in the design phase. For design space analysis and budgeting purposes, the condensed second order model as suggested by Mansuri and Kang [34] and as also used by the PCI Express Jitter modelling specification [47] will be used: The prototypical second order phase transfer function of a PLL in phase space is given as [34]

$$H_{\mathrm{PLL}}(s) = \frac{\phi_{\mathrm{out}}}{\phi_{\mathrm{in}}} = \frac{2\zeta\omega_{\mathrm{n}}s + \omega_{\mathrm{n}}^2}{s^2 + 2\zeta\omega_{\mathrm{n}}s + \omega_{\mathrm{n}}^2} \tag{2.10}$$

where ω_{n} is the PLL natural frequency and ζ the damping factor of the closed loop system. The relation between natural frequency and the 3 dB cutoff frequency of the PLL equates

to $\omega_{3dB} = \omega_n \sqrt{1 + 2\zeta^2 + \sqrt{(1 + 2\zeta^2)^2 + 1}}$. On the one hand, we do not particularly care about the exact interdependences of a detailed PLL design here. On the other hand, we would like to use the link budgeting procedure and framework to derive general target specification values for the subcomponents of the PLL and weigh their impact against other subdesign choices. Therefore, we will go a step further than the two bare performance parameters mentioned above and discuss the PLL design in some more detail (see again figure 2.6). This is also necessary to arrive at the ultimate goal of obtaining a phase noise PSD for the PLL output - the various noise sources and their contribution to the noise seen at the output need to be taken into account in this case.

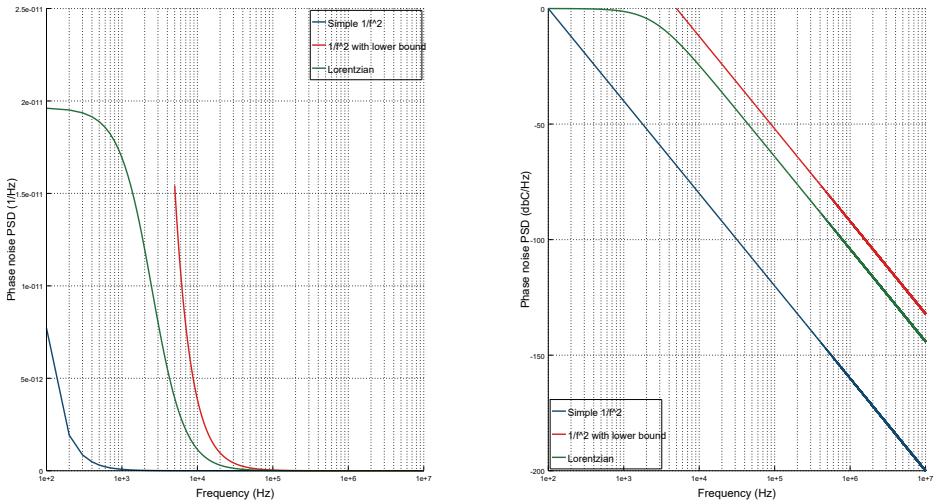

Figure 2.7: Phase noise power spectral densities of the three oscillator models presented in this text.

A PLL takes the reference clock of a quartz oscillator as its input and generates an output clock at a potentially higher frequency which is phase locked to the reference. The reference clock itself already exhibits jitter and therefore has a phase noise PSD associated with it. It is common to approximate the phase noise with

$$\mathscr{L}(f) = \frac{\sigma_{n,ref}^2 f_{ref}^3}{f^2}$$

where $\sigma_{n,ref}$ is the total RMS cycle jitter both produced (by voltage noise of internal devices) and induced (via supply) in the oscillator. This does not take 1/f noise into account. However, this noise contribution is only relevant at very low frequencies and does not much affect the performance of the serializer system as can be seen at a later point (see subsection 5.3.6). Quartz vendors do not directly provide information on $\sigma_{n,ref}$

in their datasheets. Fortunately, this is not necessary as the usual information on the total RMS output jitter of the quartz $\boldsymbol{\sigma_{\mathbf{j,ref}}}^{\ddagger}$ along with its center frequency of oscillation $\boldsymbol{f_{\mathbf{ref}}}^{\ddagger}$ are enough for the most simple model. Due to equation 2.8 the following relation for the given parameters hold:

$$\sigma_{\mathrm{j,ref}} = \frac{4}{(2\pi f_{\mathrm{ref}})^2} \int_{1}^{f_{\mathrm{j,BW}}} 2\mathscr{L}(f)df \qquad \rightarrow \qquad \sigma_{\mathrm{n,ref}} = \frac{\pi\sigma_{\mathrm{j,ref}}}{\sqrt{2f_{\mathrm{ref}}}}$$

Along with the total output jitter, vendors usually supply the frequency bandwidth $\boldsymbol{f_{\mathbf{j,BW}}}^{\ddagger}$ across which the measurement was made. For the simple $1/f^2$ model given here, this information, albeit used during the integration of the phase noise PSD, does not affect the model value $\sigma_{\mathrm{n,ref}}$. This is due to the divergent nature of the phase noise function for $f \rightarrow 0$ which is also the reason for restricting the integration above towards the lower bound to 1 Hz. This restriction of the integration interval essentially states that the bulk of noise power as given by $\sigma_{\mathrm{n,ref}}$ concentrates within 2 Hz around the carrier. This frequency precision cannot even be achieved with the best devices on the market. Some *spectral broadening* around the central carrier frequency can always be observed. If this broadening is described with the parameter $\boldsymbol{f_{\mathbf{osc},\gamma}}^{\ddagger}$ the integration above gives

$$\sigma_{\mathrm{n,ref}} = \frac{\pi\sigma_{\mathrm{j,ref}}}{\sqrt{2f_{\mathrm{ref}}}} \left(\frac{f_{\mathrm{j,BW}} f_{\mathrm{osc},\gamma}}{f_{\mathrm{j,BW}} - f_{\mathrm{osc},\gamma}} \right)$$

The rather unrealistic notion that frequencies below $f_{\mathrm{osc},\gamma}$ do not contribute to overall phase noise power puts more energy into higher frequency bands. This overestimation, however, may lead to more conservative overall designs. The error made by approximating the reference oscillator phase noise by almost Dirac like functions can be resolved by using a Lorentzian model. It has been shown that oscillators, much like many other resonating physical systems, exhibit a spectral phase noise power distribution that adheres to this form[49]:

$$\mathscr{L}(f) = \frac{c f_{\mathrm{osc}}^2 \sigma_{\mathrm{n,ref}}^2}{(\pi\, f_{\mathrm{osc}}^2\, c)^2 + f^2} \qquad \text{where} \qquad c = \frac{f_{\mathrm{osc},\gamma}}{2\pi f_{\mathrm{osc}}^2}$$

Here, for a given RMS output jitter collected over a defined frequency band, it can be shown that the parameter $\sigma_{\mathrm{n,ref}}$ can be derived from the model parameters according to

$$\sigma_{\mathrm{n,ref}} = \pi\sigma_{\mathrm{j,ref}} f_{\mathrm{osc}} \sqrt{\frac{\pi}{2\mathrm{atan}\left(\frac{f_{\mathrm{j,BW}}}{\pi f_{\mathrm{osc}}^2 c}\right)}}$$

Figure 2.7 compares the three presented models. The left picture displays phase noise magnitude and shows how the Lorentzian spectrum takes a position in between the two

simplistic models. The right hand side compares the functions on basis of the more common logarithmic representation of phase noise normalized to the respective power at the "central frequency of oscillation" (in $\frac{\mathrm{dbC}}{\mathrm{Hz}}$). Here, it becomes apparent that the simple $1/f^2$ model will produce a much too optimistic noise level due to the unrealistic spectral broadening of a few Hertz. On the other hand, the lower bound limited derivation of this model can potentially be used as a conservative estimate, especially given the fact that information about the spectral width of the central oscillation may not always be available. The phase noise of the reference oscillator undergoes a coloring process in the PLL that is given by the PLL transfer function 2.10. Its contribution to the total PLL phase noise PSD can thus be computed straight forward provided that values for natural frequency and damping factor are known. After all, this is exactly what the PLL is supposed to do: track all low frequency variations within a given bandwidth and reject the remainder which usually originates from unwanted perturbations such as electromagnetic interference or clock distribution related noise.

The natural frequency depends on the so-called open loop gain K_{loop} which in turn depends on the phase detector gain $\boldsymbol{K_{\mathbf{PD}}}^{\ddagger}$, the lowpass filter decimation gain $\boldsymbol{K_{\mathbf{LP}}}^{\ddagger}$, the gain of the voltage controlled oscillator $\boldsymbol{K_{\mathbf{XCO}}}^{\ddagger}$ and the divider feedback gain $\boldsymbol{K_{\mathbf{D}}}^{\ddagger}$ such that $\omega_{\mathrm{n}} = \sqrt{K_{\mathrm{loop}}} = \sqrt{K_{\mathrm{PD}} K_{\mathrm{LP}} K_{\mathrm{D}} K_{\mathrm{XCO}}}$. The damping factor ζ on the other hand can be shown to adhere to $\zeta = \omega_{\mathrm{n}} \frac{\tau_{\mathrm{LP}}}{2}$ where $\boldsymbol{\tau_{\mathbf{LP}}}^{\ddagger}$ is the lowpass filter time constant. All of these subcomponent parameters separate the PLL design into manageable portions and their so-defined metrics. They, of course, strongly depend on the particular, underlying design of the PLL subcomponents but can generally be derived in simulations.

The phase detector compares a divided version of the PLL output clock with the reference and produces a signal at its output that is proportional to the phase difference of the two. It, too, could either be implemented with a linear or a digital (bang-bang) approach. The resulting phase detector gain K_{pd} is more involved to analyze for the latter (as also mentioned in subsection 2.2.6 in the context of the receivers clock data recovery). The lowpass filter reduces the variability in the sequence of phase detector information and uses it to drive the voltage or digital controlled oscillator (XCO) which in turn needs to produce an input signal proportional frequency at its output. The phase detector, the divider and the lowpass filter are thought of as noiseless components. This is because either they may be built from passive devices with a narrow frequency band (low pass) and with their 1/f contribution ignored (which makes their overall noise contribution at very low frequencies negligible). The other reason may be that they are realized using all digital components (phase detector, divider and in all digital PLLs even the low pass). As a matter of fact, in recent years *all digital phase locked loops (ADPLL)* have become increasingly popular as their properties are affected favorably by technology shrinks compared to their all analog counterparts. For a more detailed treatment, refer to [38]. Therefore, the two remaining subdesigns with contribution to PLL output phase noise here are the XCO and

the PLL output buffer driving the serializer clock tree. The XCO intrinsic phase noise can be modeled just like the reference clock itself as both conceptually are free running oscillators (as long as the PLL loop is not closed). Next to the gain $\boldsymbol{K_{xco}}^{\ddagger}$, the XCO therefore also possesses parameters for its free running frequency $\boldsymbol{f_{osc,xco}}^{\ddagger}$, the associated spectral broadening $\boldsymbol{f_{osc,xco,\gamma}}^{\ddagger}$ and the total output phase noise $\boldsymbol{\sigma_{j,xco}}^{\ddagger}$ within the frequency band $\boldsymbol{f_{j,xco,BW}}^{\ddagger}$. It can be shown that the XCOs phase noise will appear at the output of the PLL under the coloring influence of the following transfer function:

$$H_{\mathrm{PLL,int}}(s) = \frac{s^2}{s^2 + 2\zeta\omega_{\mathrm{n}}s + \omega_{\mathrm{n}}^2} \tag{2.11}$$

It is the same noise transfer function that also colors the phase noise introduced by the PLL output buffer. For the PLL output buffer noise model and its resulting phase noise PSD $S_{\phi\phi,buf}(s)$, the reader is referred to subsection 2.2.5 on general buffer noise considerations. Finally, the total PLL phase noise power spectral density can be calculated according to [34]

$$S_{\phi\phi,pll}(s) = S_{\phi\phi,ref}(s)\,|H_{\mathrm{PLL}}(s)|^2 + S_{\phi\phi,xco}(s)\,\left|H_{\mathrm{PLL,int}}(s)\right|^2 + S_{\phi\phi,buf}(s)\,\left|H_{\mathrm{PLL,int}}(s)\right|^2$$

Figure 2.9 shows a subcomponent noise PSD breakdown together with the total output noise PSD of the PLL. This information or its Fourier transform, the autocorrelation function of timing error $R_{\phi\phi}(t)$, is then used as input information to the link budgeting procedures described in chapter 5.

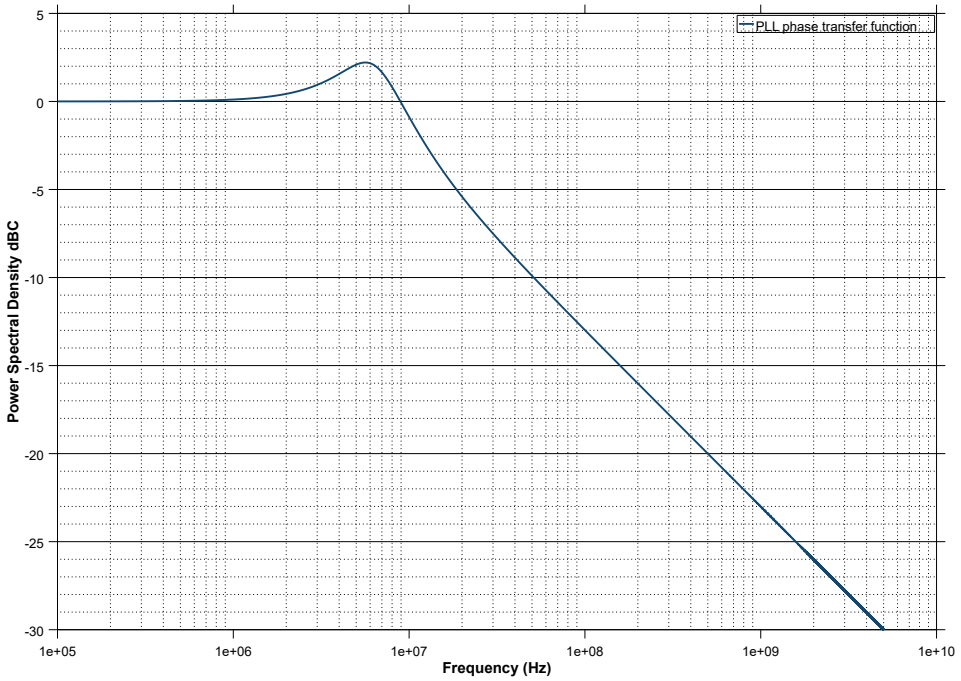

Figure 2.8: Example of a PLL phase transfer function with $10.9\,\mathrm{MHz}$ bandwidth and relatively strong peaking with which the phase noise spectral densities below were generated.

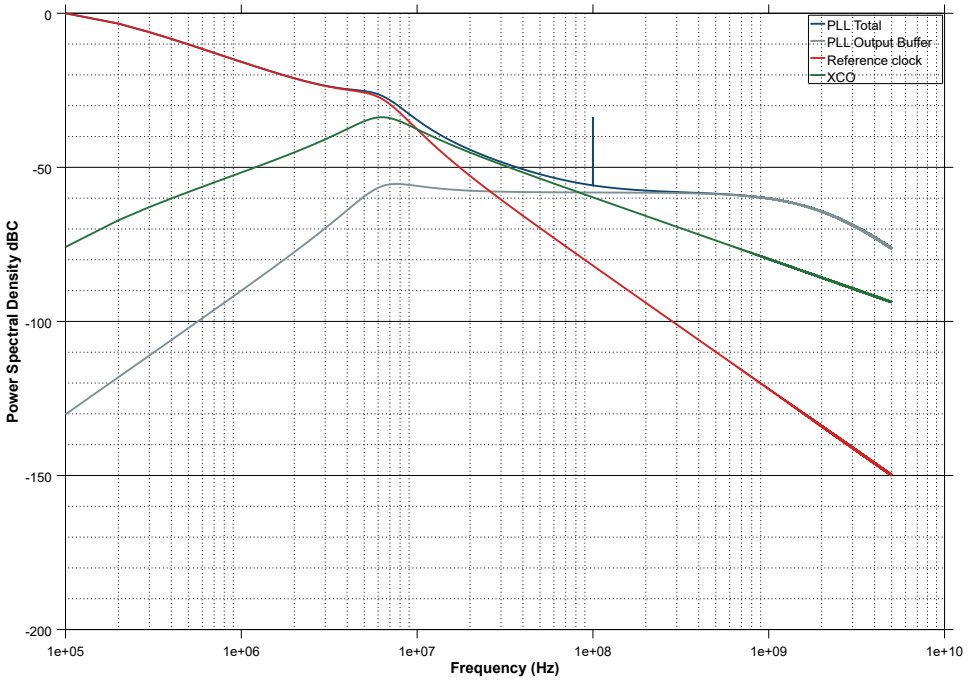

Figure 2.9: Phase noise power spectral densities of the PLL and its subcomponents. The total output jitter across the entire band up to $5\,\mathrm{GHz}$ is $690\,\mathrm{fs}$.

2.2.5 Transmitter

The transmitter accepts parallel data at its digital interface from the chip domain and serializes the data onto the transmission line. The product of clock frequency and data width must stay constant unless a transition in data coding is being made. The most simplistic example of this would be the transition from a single to a double data rate NRZ coding scheme. The general transmitter architecture, just like the general topology of the PLL, can be well described structurally with hardware description languages (see section 4.2). This includes for example the choice of logic mode, muxing by passgates or by logic gates or the overall clocking scheme. With the flexibility of real number modeling as provided by VerilogAMS and SystemVerilog, the output buffers themselves, too, can be described abstractly with respect to their most important performance metrics (see again section 4.2).

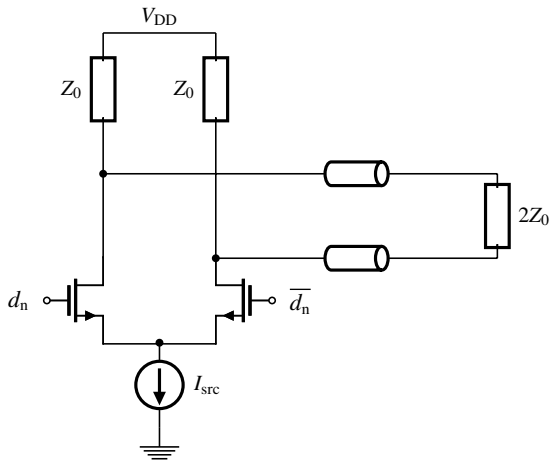

Figure 2.10: CML transmitter

Transmitter output buffer architecture was mainly focused around current mode logic (CML) implementations in the early days of multigigabit transceiver designs. CML features a switched constant current source source whose output is fed into two pullup resistors matching the impedance of the channel thereby producing the required output voltage (see figure 2.10). Its major drawback is the portion of constant current between supply and ground potential that does not flow through the receivers termination resistance. A major benefit, however, is its inherent linearity and the fairly easy generation of weighed input signals by current summation as required for a feed forward equalizer or other more complex amplitude modulation schemes. In recent years, source series terminated output drivers have become more popular (see figure 2.11)[35][25] even with more advanced modulation schemes[28]. The power benefit without clocking overhead of SST architectures is about a factor of four for the same output, waveguide and termination

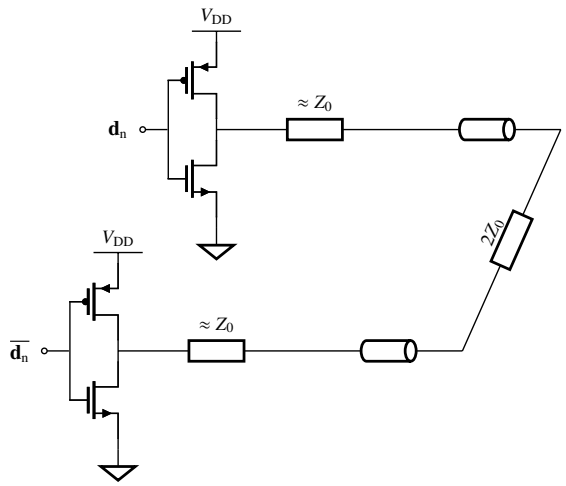

Figure 2.11: SST transmitter

impedance. A further power benefit can be achieved, if transmitter and receiver alike are designed to operate single-ended instead of differentially[50]. These setups, however, require good control of current return paths across the power planes and are less noise immune which makes them appropriate mainly for short-haul applications.

The design tradeoff between CML and SST is an increase in overall complexity and more stringent requirements on duty cycle distortion control, output impedance adjustment circuitry and linearity analysis. Also, due to the reduction of supply voltage in more advanced technology nodes, it becomes ever more difficult to realize the output amplitudes required to support more sophisticated modulation schemes. Remedies to this circumstance have been conceived in the past ranging from thick oxide output stages to stacked and capacitively coupled thin oxide drivers [35]. CML output stages, on the other hand, suffer from severe clock feedthrough in advanced technology nodes when combining output stage and last multiplexer stage. This stems from the increasing size of the gate drain capacitance with respect to the gate source capacitance of the transistors in these technology nodes. The details are again beyond the scope of the discussion here.

For a more general analysis, the transmitter architecture is thought of being a voltage mode digital to analog converter with a preceding weighing matrix such that the input symbol stream is buffered and converted to a digital representation of the modulation strength required at any given point in time. Relevant metrics for its characterization are the output impedance (tuning range) $\boldsymbol{Z_{TX}}^{\ddagger}$, the total output signal swing $\boldsymbol{V_{tx,pp}}^{\ddagger}$ and the output signal rise and fall times $\boldsymbol{t_{r,tx}}^{\ddagger}$ and $\boldsymbol{t_{f,tx}}^{\ddagger}$ (especially their asymmetry which will result in output common mode variations). The termination and electrostatic discharge protection (ESD) compensation quality could also be included as a part of the channel but is attributed to the transmitter here.

As previously discussed, the combination of output rise and fall times, power supply rejection ratio, power supply noise and internal noise sources will jointly result in the signal

phase and voltage noise added by the transmitter stage. This phase noise is in addition to that introduced by the clock distribution and PLL which are directly fed to the transmitter output due to its inherent retiming functionality. The voltage noise sets the initial SNR at the transmitter output. However, since the transmitter is a quasi periodic system at best (if user data does indeed exhibit periodicity such as is the case for PRBS patterns), the phase noise would not be the proper metric to solely rely on. As described in section 2.2.1, the voltage noise produced by a transmitter will eventually be mapped to phase noise at the sampler input by virtue of the finite signal rise and fall times at this point. It is therefore possible to handle timing and voltage errors independent of one another if carried out stringently. The budgeting procedure presented in this text will rely on this distinction (see section 5.3). The model parameters used for the transmitter output buffer are basically the same as for other amplifier and buffer stages in this text. The influence of the PDN on total output noise of a buffer is described by its power supply rejection ratio (PSRR). Once actual implementations are available, this information can generally be simulated. However, the frequency dependent magnitudes strongly depend on device mismatches in the given design. Although static offsets and mismatches can be counterbalanced with appropriate schemes, the transfer function from supply input to buffer output may still be changed (degraded) with respect to the perfectly symmetric case (all high speed design entities are thought of being differential in this text). A Monte Carlo analysis would then need to find the worst case PSRR. Due to this overly complex circumstance, the PSRR buffer information $\boldsymbol{H}_{\mathbf{vdd2o}}(\boldsymbol{\omega})^{\ddagger}$ is considered to be a worst case mask rather than an exact function. Similiar considerations apply to the buffer intrinsic noise profile $\boldsymbol{S}_{\mathbf{VV,int}}(\boldsymbol{\omega})^{\ddagger}$ which can be modelled generically but should always be considered more of an upper bound constraint than an exact representation of the in-band noise produced by the subcomponent. In the most simple case, $\boldsymbol{S}_{\mathbf{VV}}(\boldsymbol{\omega})^{\ddagger}$ is modelled as a colored noise source with its first order pole given by the output pole $\boldsymbol{\omega}_{\mathbf{3db}}{}^{\ddagger}$ of the buffer itself and an estimate of the total RMS voltage noise $\boldsymbol{\sigma}_{\mathbf{n,int}}{}^{\ddagger}$ within this frequency band (see equation 2.9). In cases where the output voltage noise power spectral density needs to be converted to a phase noise spectral density, the buffer signal rise and fall times $\boldsymbol{t}_{\mathbf{r,buf}}{}^{\ddagger}$ and $\boldsymbol{t}_{\mathbf{f,buf}}{}^{\ddagger}$ need also be known for the given, periodic pattern. In these cases, instead of simulating the PSRR and noise of the given buffer, a periodic steady state analysis in conjunction with a periodic transfer function analysis for the supply interaction and a periodic noise analysis for the output phase noise due to device internal noise sources could also be carried out. This is especially helpful for the clock distribution where long chains of buffers of potentially different logic types need to be accounted for. This reduces the amount of data to be processed for the transmitter clock tree to two input metrics for budgeting: $\boldsymbol{S}_{\boldsymbol{\phi\phi}\mathbf{,tx,clk}}(\boldsymbol{\omega})^{\ddagger}$ for intrinsically produced phase noise and $\boldsymbol{H}_{\mathbf{vdd2,tx}\boldsymbol{\phi}}(\boldsymbol{\omega})^{\ddagger}$ for its power supply sensitivity.

The transmitter of the system examined in this text features a finite impulse response (FIR) for preequalization of the data stream. Details on how this affects the overall system

response and further implications are presented in the next section.

2.2.6 Receiver

Figure 2.1 also shows the most budgeting relevant subcomponents of the receiver. As was the case with the transmitter, this does not include the receiver termination and ESD compensation, the all digital deserialization stage or the (very complex) clocking scheme. A more detailed picture of transmitter and receiver can be found in chapter 6. For budgeting analysis, however, the clocking scheme again boils down to the two metrics $\boldsymbol{S}_{\phi\phi,\mathbf{rx,clk}}(\omega)^{\ddagger}$ and $\boldsymbol{H}_{\mathbf{vdd2,rx}\phi}(\omega)^{\ddagger}$. While the transmitter did not act on this information at all, the receiver must process its spectral content in a much more complex way. The reason for this lies in the clock data recovery (CDR) circuitry that is used to extract the very clock from the data stream with which it was put on the transmission line (see below). This is also the reason for the increase in clock distribution complexity. There must be a subcomponent in place that allows to phase shift the receivers sampling clock at a level of reasonable granularity. In the past, all analog solutions based on voltage controlled oscillators (VCO) have been used for this purpose and it is no coincidence that analysis and metrics of clock data recovery resemble that of PLLs somewhat - a circumstance this text aims to exploit for budgeting at a later point (see section 5.3). As PLL designs transition to more digital implementations for future technology nodes, CDR designs follow suit. The role of the VCO as a voltage to frequency/phase translation unit is replaced by a digital to phase converter with the phase interpolator being the most popular implementation (see below). Before clock recovery can be attempted, however, the incoming data signal needs to be preprocessed. In broadband communication systems, the major equalization effort is usually performed in the receiving side. This includes the time continous equalization stages (CTLE), oftentimes variable gain amplifiers (VGA) and, especially in the context of backplane channels, decision feedback equalization (DFE). For more information on equalization and its effect on overall system response, the reader is again referred to the next chapter.

One of the most central subcomponents of the receiver is the sampler (actually the bank of samplers). Here, the analog, equalized but noise impaired input signal is converted back to a digital representation. Depending on the line coding scheme, the number of available clock phases and the CDR phase detector design (see below), the receiver architecture may require a substantial amount of samplers. For a detailed discussion on the sampler model used in this text, refer to section 4.5. The overall architecture of the receiver can again be described very well with a structural, hardware description language based approach. Deterministic effects of the architecture such as residual offsets after calibration, duty cycle distortion, the comparably large time constants of the CDR (see below) or the effects of residual intersymbol interference (see next section) can thus be analyzed and accounted

for very well.

However, the most central aspect of clock recovery must be treated either statistically (Markov chains) or with phase space, linearized models. The complexity and far reaching consequences of this particular subcircuit deserves a more detailed treatment here:

2.2.6.1 Clock data recovery circuit

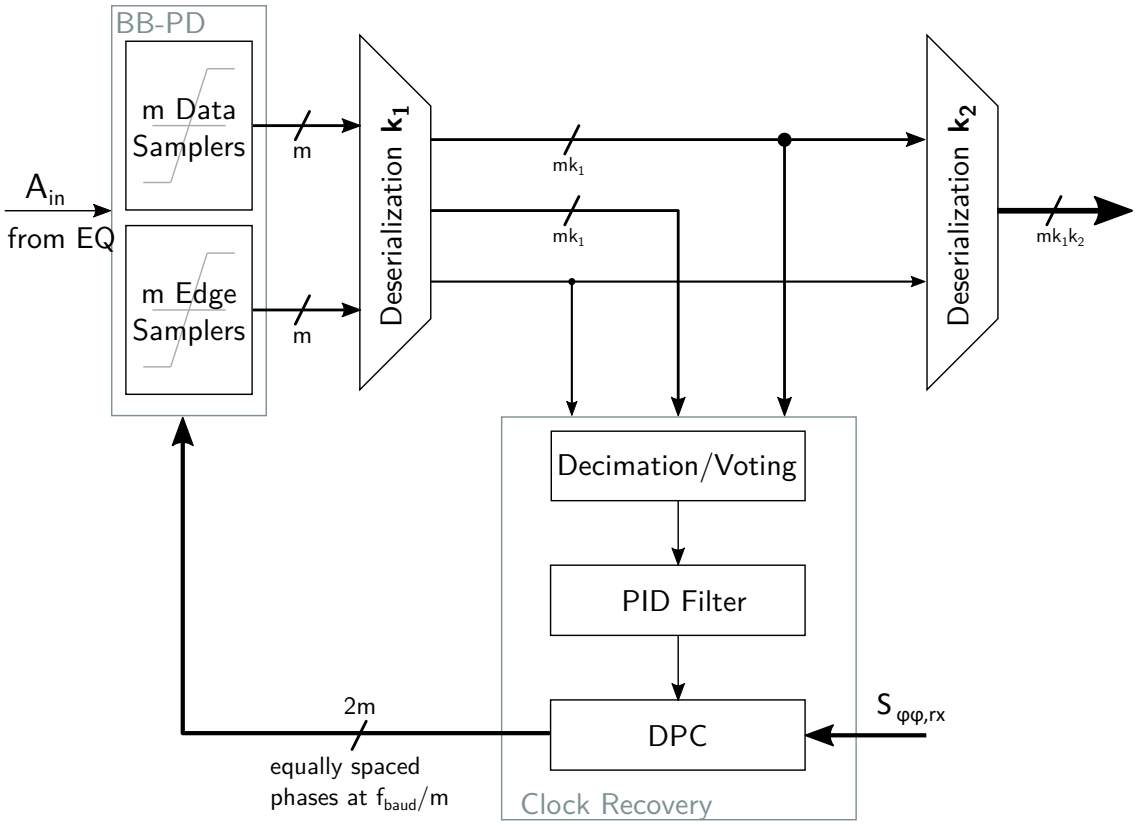

Figure 2.12: A CDR based receiver under omission of termination and equalization. The two banks of samplers form the analog to digital conversion stage (data samplers) as well as the phase detector (data and edge samplers). Note that the number of samplers may increase by a factor of n if an n-way speculative DFE is implemented with the receiver (see section 3.3.3) PAM-N schemes increase the number of required samplers even further.

Figure 2.12 shows yet another, less system budgeting centric view of a CDR based receiver architecture. The preconditioned analog, full rate signal is distributed to all samplers. There are *data samplers* for analog to digital conversion as well as so-called *edge samplers* which form an all-digital (bang-bang) phase detector (BB-PD). These types of phase detectors are highly non-linear and need to be analyzed thoroughly (see below). As is the case with PLLs, there have been CDR architectures relying on linear phase detection schemes [23] such as analog mixer based CDR circuits. They allow the

extraction of the phase detectors key parameter, its gain $\boldsymbol{K_{PD}}^{\ddagger}$, more directly than their all-digital counterpart. Irrespective of how the actual phase detector is implemented, the information at its output represents the phase offset between the local receiver clock and the clock embedded in the incoming digital data stream.

All-analog CDRs are in close resemblance to analog PLLs and pass on the analog phase detector output to a filter which in turn steers the frequency of a voltage controlled oscillator (VCO). From the VCO, all the clock phases needed for the samplers would be derived directly. A (filtered) variation in voltage due to the detected offset would therefore lead to a variation in the sampling time instant of the receiver. Since a receiver would need to track both phase and frequency offsets of the incoming data (after all, the reference oscillators and PLLs of trasmitter and receiver are usually not identical), the filter needs to have an integral (frequency tracking) as well as a proportional (phase tracking) component.

With all digital CDRs (as they are discussed here), the basic recovery procedure is essentially the same. With the output of the phase detector being a digital signal, the filter, too will operate on entirely digital information. Accumulators can be used for performing integration while a simple shift operation is sufficient to realize proportional gain. Digital vectors can be used to adjust the proprtional $\boldsymbol{K_P}^{\ddagger}$ and integral gain $\boldsymbol{K_I}^{\ddagger}$ for optimum operation. An HDL based description of the signal processing part of the CDR is thus possible and even recommended as it takes care of both the modeling and implementation with standard industry flows at the same time. However, even the most advanced technology nodes only allow semi-custom implementation up to a few GHz at best. This is why usually, the CDR must be located after a deserialization stage thus running at lower frequencies but on more data in parallel. This intermediate step of deserialization introduces a delay in the regulation loop of the CDR - $\boldsymbol{N_{dly}}^{\ddagger}$. Since there is a binary phase detector decision for every bit received, there must be a so-called *decimation stage* that allows to extract an average phase information from this stream of incoming data. Needless to say that decimation will thus have an impact on the overall gain in the CDR filter loop and increase the delay of said loop further. Numerous techniques have been proposed for decimation including boxcar filtering and equal [58] or unequal majority voting [67]. For the discussion here, we note that the central parameter $\boldsymbol{K_D}^{\ddagger}$ - the *decimation gain* - can be extracted from all digital testbenches (see [67]) and its magnitude can thus be examined or constrained during analysis and budgeting. The output of the digital filter then needs to be converted to a particular clock phase at the samplers. For this purpose, delay locked loops have been used in the past. Their power consumption becomes an increasing problem as data rates grow and their accuracy suffers greatly from the larger physical variations at more advanced technology nodes. A more robust and widely used approach is the so-called *phase interpolator*. For a thorough description the reader is again referred to the available literature [11, 38]. The key parameter of a phase interpolator in the context of the CDR control loop is its digital to phase converter (DPC) gain $\boldsymbol{K_{DPC}}^{\ddagger}$. K_{DPC} is just given by the

number of steps per radian (or bit time UI) that can be performed.

There a two well-established ways in which CDR circuits can be analyzed. The first is to treat the CDR statistically by a Markov chain analysis. This analysis is the preferred choice in literature about general communication research [60] and takes into account most non-linear effects. It is also the way in which some contemporary link budgeting procedures treat the CDR [60][44].

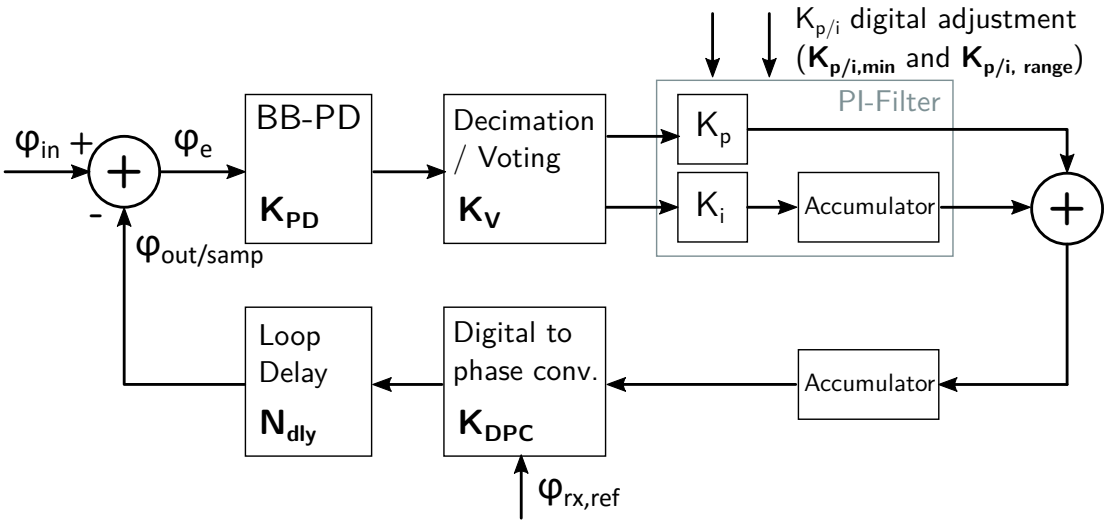

Figure 2.13: The (linearized) control loop view of the clock data recovery circuit with annotated budgeting parameters

The alternative is to describe the CDR system in phase space. For this to work, all of its components, even the heavily non-linear BB-PD, have to be linearized. The resulting control loop system can be seen in figure 2.13. The linearization will constrain the validity of the model to well-equalized systems with moderate jitter magnitudes (see section 5.3 for details). Its benefit, however, is its general applicability to phase noise spectral density analysis (again, see section 5.3) and the straight forward derivation of the relevant budgeting and specification metrics - the *jitter tolerance* $H_{\mathrm{CDR,jtol}}(s)$ and sometimes the jitter transfer characteristic $H_{\mathrm{CDR,jxfer}}(s)$. While the former describes how much input jitter present at a particular frequency (thus: the magnitude of the phase noise PSD at the given frequency) can be tracked by the CDR loop, the latter is a measure on how much jitter is passed on to subsequent components in the communication chain if the recovered clock is used and no retiming occurs. This is primarily important for transceiver designs which operate as a repeater stage in very long physical communication channels.

Two characteristic functions for CDR performance characterization, the jitter tolerance

and jitter transfer functions, can be derived from the CDR loop gain [58]:

$$H_{\mathrm{CDR,loop}}(z) = \left(\frac{K_{\mathrm{PD}}K_{\mathrm{D}}K_{\mathrm{DPC}}}{1-z^{-1}}\right)\left(K_{\mathrm{P}} + \frac{K_{\mathrm{I}}}{1-z^{-1}}\right)z^{-N_{\mathrm{dly}}} \tag{2.12}$$

Especially the delay factor $z^{-N_{\mathrm{dly}}}$ makes an analytical treatment of this problem quite impossible. Numerical tools and the bilinear transformation can be used to derive the loop gain function in its Laplace form $H_{\mathrm{CDR,loop}}(s)$ which we can then use to compute the jitter tolerance function to

$$H_{\mathrm{CDR,jtol}}(s) = \left(1 - \frac{k\sigma_{\mathrm{j,rx,s}}}{T}\right) \cdot (1 + H_{\mathrm{CDR,loop}}(s)) \tag{2.13}$$

where $\sigma_{\mathrm{j,rx,s}}$ is the total RMS jitter seen at the receiver sampler, $k = 2SNR(BER)$ is twice the signal to noise ratio required at the target BER level and T is, as usual, the bittime.

The jitter transfer function on the other hand is given by

$$H_{\mathrm{CDR,jxfer}}(s) = \frac{H_{\mathrm{CDR,loop}}(s)}{1 + H_{\mathrm{CDR,loop}}(s)} \tag{2.14}$$

These functions do not take into account that the local oscillator itself is jittered, too, and that the clock distribution of the receiver (which the phase interpolator(s) is obviously a part of) adds further phase noise to the sampling clock as well.

How this circumstance is considered during budgeting together with more details on the validity of linearization can be found in section 5.3. To conclude this section, a final remark on a common metric to describe the performance of a receiver - the *receiver sensitivity* - has to be made. Again, this is a metric from the rather narrow band radio and microwave frequency domain. It is defined as the minimal signal strength at the receiver input that still leads to an error free operation within the given BER limit:

$$S_{\mathrm{min}} = (S/N)_{\mathrm{min}} k_{\mathrm{B}} T_0 \int_0^B NF(f)df$$

where $(S/N)_{\mathrm{min}}$ is the minimum required signal-to-noise ratio to detect the signal at a given BER, k_{B} is the Boltzmann Constant, T_0 the nominal temperature at receiver input (290 deg K), B the bandwidth of receiver and $NF(f)$ the frequency dependent noise figure of the receiver.

As described in subsection 2.2.1, this view on sensitivity, which has a voltage signal to noise ratio in mind, is only a fraction of the information required - in this case the height of the signal eye at the sampling instant as chosen by the receiver clock recovery circuit. There are, however, more constraints to the operation of a wideband receiver such as the phase noise spectral density of the incoming signal. Also, the residual equalization error of

the signal at the sampler inputs and the residual receiver offset is of importance (a difficult task once non-LTI equlization schemes such as a decision feedback equalizer (DFE) must be included, too). All of these system imperfections, next to being detrimental to the input signal itself, lead to a reduction in phase detector gain which in turn degrades the CDR loop bandwidth and thus its capability to track jitter beyond this cutoff effectively. The result is a decrease in eye width which may disallow operation at the target BER level. It is due to these very complicated interactions in a system comprised of a sizable amount of subcomponents that an automated and tightly integrated modeling and budgeting mechanism is of the essence, especially as data rates continue to grow and feature sizes continue to shrink (with thereby growing subcomponent variations). Nevertheless, the receiver sensitivity metric is of value when channel and transmitter which provide the reference input signal are also known along with the properties discussed in the foregoing paragraphs.

For structural reasons, this text has only hinted at the importance of the transmission channel and the equalization stages in the serializer system so far. The next section is entirely dedicated to this subject and highlights the fact that it is actually the range of supported physical channels that puts the most severe constraints on overall system architecture and the final performance metric, the bit error rate.

3 Electrical transmission channels and equalization

The most central aspects of a serializer based communication link have thus far been omitted in the course of the system description: The physical communication channel with its termination scheme as well as the equalization features in transmitter and receiver to compensate its nonidealities. The performance and power efficiency targets of modern serializer technology might not be solvable by technology scaling or circuit improvements alone. Properties of the transmission channels, most notably impedance and allowable length, need to be revised as well. In order to get a more thorough understanding as to how transmission channel properties affect serializer system performance (and via provision of potentially required equalization stages also power efficiency), a closer look at the realms of electrical communication is required. The ultimate goal is a parameterizable channel model which together with the analysis and budgeting framework presented here allows numerical investigation into performance (and in the future even power) scaling of serializers with a broad range of equalization options.

From a quiet general point of view and as far as the analyses in this text are concerned, transmission lines will either be described with their S-Parameter matrices or its Fourier transform, the impulse response, from one port to another. A *port* is the combination of one end of a signal conductor with its (shield and signal) ground conductor. A single coaxial cable therefore has two ports, a differential signal pair of a PCB four. S-Parameter matrices (sometimes also called *Touchstone files*) are the set of all frequency transfer functions between every two ports of a channel for a fixed source and load termination impedance matched to the *electromagnetic wave impedance* Z_0 of the channel [51].

Sometimes, there is only a single bit response (SBR) given in which case the impulse response must be retrieved by deconvolution with the appropriate pulse function. The single bit response view is very useful when either a good visual measure of equalization success is needed (see section 3.3) or a statistical assessment algorithm on potential channel performance is required (see also 5.2).

In most cases, there will be no distinction here between single ended and fully differential channels. The latter are more involved to described but can be decently abstracted with so-called *mixed mode S-Parameters* [3] which are basically a set of linear transformations combining the ports of the signal traces that form the differential pair. Figure 3.5 gives

an example of a channel described by both its forward transfer function (from one end of the cable to the other) and the associated SBR. The SBR in figure 3.5 is normalized in time to the unit interval T of the bit pulse with which it was created. Ideally, there would only be a signal unequal to zero at the so-called *main cursor* $t = 0 \cdot T$. However, due to the dispersive (group delay) and lossy properties of the channel, there is a residual signal at time instants $t = n \cdot T$. Other bits which are transmitted at these instants in time have their main cursor voltage superimposed with these perturbations which is why this effect is called intersymbol interference (ISI).

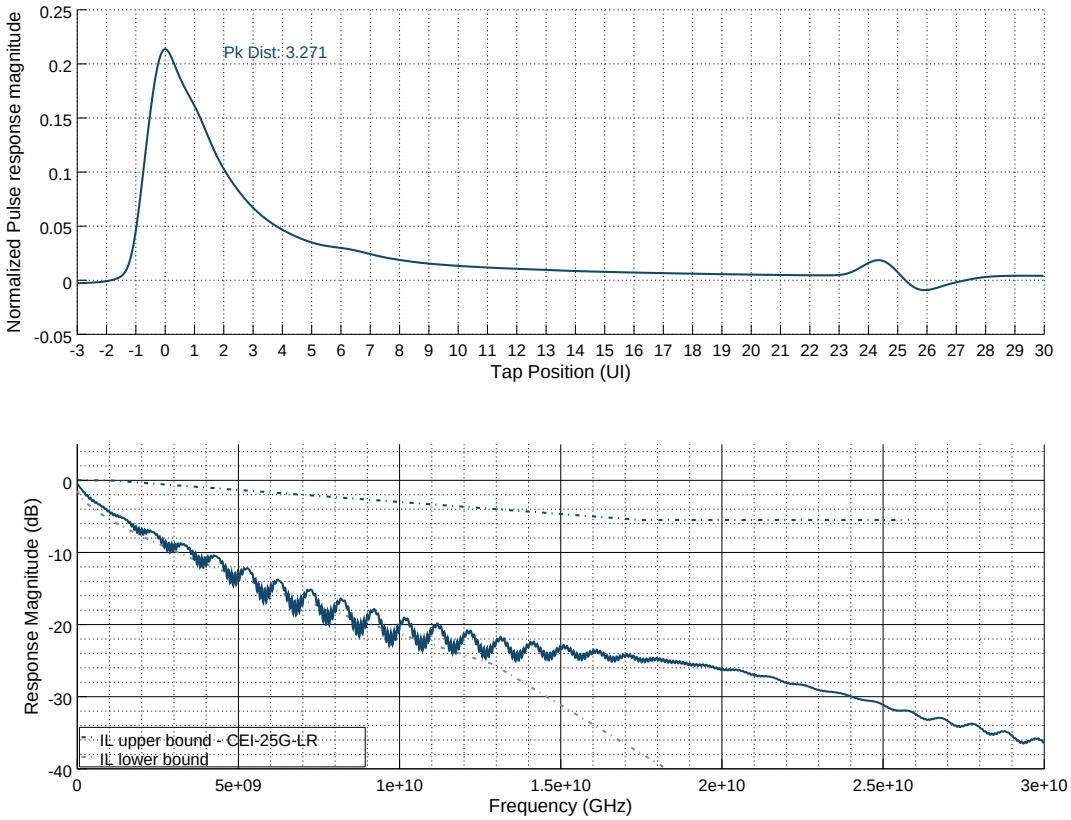

Figure 3.1: SBR at UI=40 ps and frequency response of a 25 Gigabit worst case CEI-25G-LR composite channel from ECAD generated S-Parameter data (compliance mask shown in dashed lines)

As can be guessed from figure 3.1 already, a closed form equation to describe the frequency transfer characteristic is very hard to find analytically. Oftentimes and especially at higher frequencies, the connectors, landing pads of coupling capacitances or even the

package balls of a chip or die pose enough of a physical discontinuity to severely change the wave impedance for a small section of the physical channel. At the spatial location of the impedance discontinuities, a part of the incident signal power of the electromagnetic wave is reflected. Its magnitude depends on the magnitude of impedance mismatch and the scattered electromagnetic wave then travels down the transmission line in opposite direction. This is also one of the reasons for using source end termination and why it has to be matched to the wave guide impedance: the termination serves as an absorber for backreflected electromagnetic waves. Signal power which is reflected $2N$ times within the channel and which is thus travelling towards the receiver once more, must naturally travel a longer distance to arrive at the receiver input. This portion of the signal reveals itself as additional ISI at tap locations very far away from the main cursor (see tap 24 and 25 of the SBR in figure 3.1). Pairs of discontinuities may form a filter for particular wavelengths depending on their spatial separation and the wave propagation velocity in between them. They reflect in the sporadic, periodically spaced dips of the transfer function as can also be seen in figure 3.1).

Instead of analyzing a transmission channel in terms of its frequency response by using a vector network analyzer for instance, one can also use a time domain reflectometer (TDR). It reveals the location of discontinuities within a transmission channel by sending a step function of well defined rise-time into the channel under test. It then records the signal waveform incident on its own output and recomputes the waveguide impedance in relation to the time passed since sending out the initial step. Figure 3.2 shows the result of such a measurement for the channel whose frequency response and SBR are given in figure 3.1. A commercial ECAD tool was used for this purpose which mimics the TDR process numerically. One can see that in order to obtain the discontinuities, the channel was designed to feature different sections of waveguides as they would also appear in a typical HPC network link for instance (compare figure 2.4 of section 2.2.3). All sections themselves (except for the discontinuities) have a differential impedance of $100\,\Omega$. There is a section of package trace right after the chip followed by the discontinuity due to the cross over from package to PCB with landing pad and chip balls (assuming a flip chip arrangement). The very small traces of the package exhibit a strong skin-effect and conductor loss (see below) and thus lead to a sharp increase in computed impedance. The strong discontinuity makes the PCB waveguide section appear a lot more low ohmic than it really is. There is a further discontinuity from PCB to the AWG interconnect cable due to the cable connector and its landing pads. The middle of the channel then is the symmetry plane of the interconnect.

In essence, while for general considerations and design space exploration simple channel models based on analytic equations are perfectly suitable, serializer verification and budgeting frameworks need to work with true S-Parameter based channels, too, in order to capture the various effects that may be detrimental to signal quality at the receiver input.

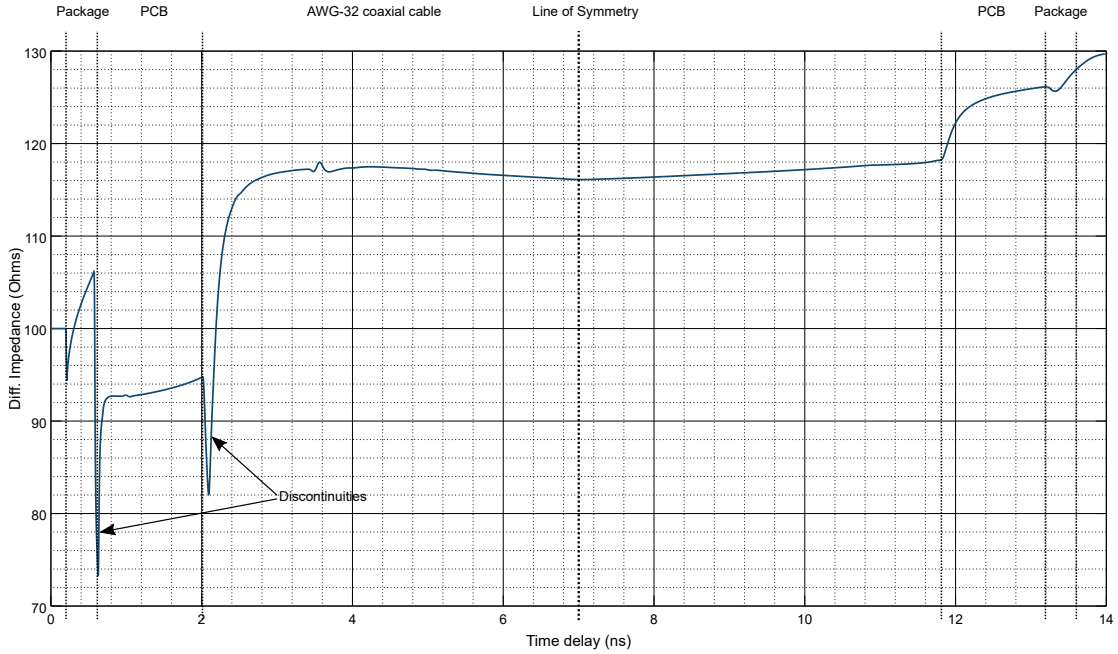

Figure 3.2: TDR simulation of the transmission channel with transfer function as given in figure 3.1 exhibiting various discontinuities responsible for the resonances

Examples of toplevel design space exploration profiting from a simple numeric channel model are given by papers which deal with feasibility and energy efficiency of exascale HPC systems (such as [29] and [59]). They usually use the so-called *bit rate capacity* of a physical, electrical transmission channel to express and predict I/O performance in terms of system size and integration density, i.e., to arrive at scaling laws for area cost of communication. An introduction on how this metric is derived will be given below along with an explanation of its particular shortcomings and problems when it comes to estimating the performance of modern serializer based communication schemes. This motivates the introduction of the Johnson signal model for coaxial transmission lines as described thereafter. This model can then already be used to derive bit rate capacity scaling laws for comparison with previous work, even without an actual serializer system model.

3.1 The bitrate capacity

In a paper from 1997, Miller and Ozaktas introduce the concept of a so called bitrate capacity of a channel [36] which is defined to be the upper limit of data transmission capability when no equalization or signal refresh is applied. By starting with a coaxial line (due to its relative mathematical simplicity) and by arguing that scaling other transmission channel types such as microstrip and stripline geometries would basically adhere to the same kind of physical constraints for reasonable sizing, they derive a general equation for

the *bitrate capacity*

$$B = B_0 \frac{A}{L^2} \tag{3.1}$$

where

$$B_0 = \frac{1}{50} \frac{16\pi Z_0^2 \sigma r_\mathrm{i}^2}{\mu r_\mathrm{o}^2} \tag{3.2}$$

with inner conductor radius r_i, outer conductor radius r_o, waveguide impedance Z_0 as well as the electric conductivity σ and the magnetic permeability μ of the conductor material. The central aspects of linear upscaling with channel bisection area and quadratic decrease with transmission line length are a consequence of restricting the analysis to the so-called *RC and skin-effect limited frequency regions* of the channel. Several other assumptions are required to obtain the constant B_0 for a particular type of channel among which there are the already mentioned absence of equalization and repeating, the constraint to NRZ line coding as well as the randomly chosen receiver detection threshold which assumes a large SNR requirement at the receiver for a somewhat all-digital based communication scheme. As already discussed in section 2.2.1, bit reception is however subject to both voltage and time domain SNR constraints as data rates enter the multigigabit domain. Noise properties of a system therefore need to be taken thoroughly into account.

Essentially, while B_0 may be rather arbitrary due to the reasons above, the general scaling with cross section area and length within the mentioned frequency regions already leads to interesting constraints for HPC systems. First, the scaling law forces stronger localization for higher data rates (a well observed trend in modern architectures, especially when moving the dynamic memory closer to the processing units for better throughput). Next, RC wirelines on chip may not resolve the problem as they grow longer [59] (although their impedance domain and termination may be more favorable).

The goal of this text is to establish a framework to analyze serializer systems with channels that are either given by specification constraints or by advanced models comparable to those used for deriving the bitrate capacity. From a design space exploration perspective for network links in the context of HPC systems this is most desirable. The design and budgeting framework would benefit from a concise, analytically derivable and numerically tractable channel model. A good candidate from literature is presented in the following. Compared to the simplified assumptions of the bitrate capacity derivation, the model covers a wide range of aspects to signal integrity especially towards higher Gigahertz frequencies and thus allows a more accurate derivation of the functional relationship between the parameters of a transmission line and its potential performance limit.

3.2 Performance regions and the Johnson Signal Model

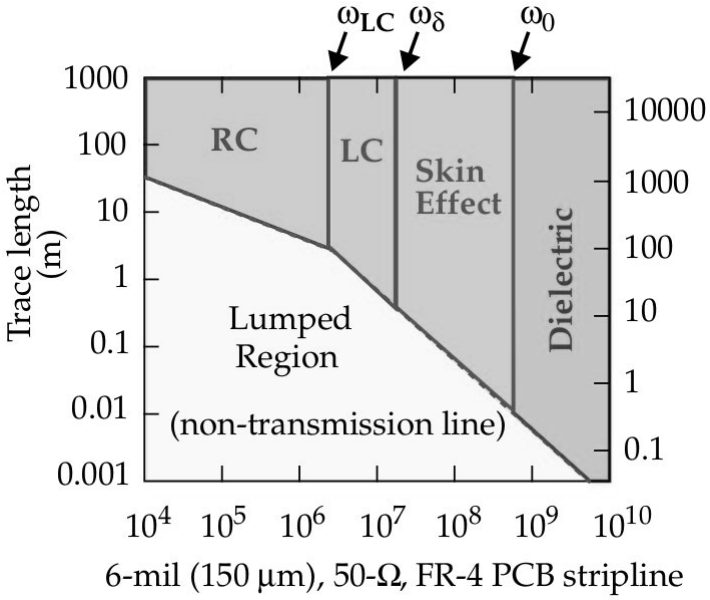

Figure 3.3: Electrical transmission line performance regions for a stripline trace on FR-4 laminate (from [24]) without waveguide dispersion region

As mentioned above, transmission lines possess various regimes of operation that Johnson [24] has dubbed *performance regions* and has provided formulas and boundaries for. The extends of a region depend on the physical realization and length of the transmission line and the range of frequencies that are to be transmitted across them. The definition of bitrate capacity assumes the physical channel in question to be within the regime of the skin-effect or RC limited regions. Here, a brief overview of all regions and their boundaries is given to highlight in which domains multigigabit wideband transmission actually takes place. Since we will be especially interested in verifying or ammending the assumptions used thus far in HPC technology predictions, an emphasis will be put on underlining the signal attenuation properties (thus, the transfer function) of a particular region. It must be noted though, that the exact transfer function is subject to the termination at source and load. However, the general relationship between attenuation and frequency (i.e. the assertion "attenuation in dB scales linearily with frequency") is unaffected by it. As in the context of the serializer system presented here, both source (transmitter) and load (receiver) terminate the transmission line with its characteristic impedance, this important detail will henceforth be ignored. At the end of this subsection, the Johnson signal model for a coaxial line is introduced. It will be used in chapter 6 to numerically derive proportionalities between feasible data rate and channel parameters such as those stated by equation 3.1. The model can be broken down into its physical and its implementation

agnostic set of equations. Naturally, the specific physical implementation will yield unique quantities of unit- capacitance, inductance and resistance. However, Miller has argued that the impedance Z_0, once set, constraints all feasible physical realizations to per-unit quantities comparable to those of the coaxial line [36] which renders it a good starting point for later analysis efforts.

Quite generally, a transmission line can be described by its characteristic impedance $Z_C(\omega)$ its propagation coefficient $\gamma(\omega)$ as well as by its propagation function $H(\omega, l)$ (unterminated condition) where

$$Z_C(\omega) = \sqrt{\frac{j\omega L_0 + R(\omega)}{j\omega C(\omega)}} \tag{3.3}$$

$$\gamma(\omega) = \sqrt{(j\omega L_0 + R(\omega)) \cdot (j\omega C(\omega))} \tag{3.4}$$

$$H(\omega, l) = e^{-l\cdot\gamma(\omega)} \tag{3.5}$$

Here, L is the series inductance per unit length and C is the shunt capacitance per unit length of the transmission line. Also, it can be shown that the overall system gain (including source and load termination) can be calculated to

$$G = \frac{1}{\left(\frac{H^{-1}+H}{2}\right)\left(1 + \frac{Z_S}{Z_L}\right) + \left(\frac{H^{-1}-H}{2}\right)\left(\frac{Z_S}{Z_C} + \frac{Z_C}{Z_L}\right)}$$

with the source and load terminations $Z_{S/L}$. Assuming that both source and load termination are purely resistive and therefore independent of frequency, attenuation will depend solely on both the characteristic impedance and the propagation coefficient as a function of frequency.

For a transmission line with a wave impedance of Z_0 at a chosen frequency ω_0 (usually set to the bandwidth requirement of the system or to a value for which the exact values of dielectric constant and its loss tangent are known, see below), the per-unit inductance of the transmission line L_0 is assumed to be fixed and all changes to series impedance with frequency is attributed to the term $R(\omega)$. Per-unit inductance L_0 and per-unit capacitance C_0 are a consequence and hence derivable from the particular physical implementation and serve, among others, as the variables of abstraction here. Apart from the wave impedance $Z_0 = \sqrt{L_0/C_0}$ for an electromagnetic wave at a particular angular frequency ω_0, the wave propagation velocity can be derived to $v_0 = \frac{1}{\sqrt{L_0 C_0}}$. The frequency dependence in the series resistance originates from the skin-effect to the most part (see below). The DC value on the other hand originates from the finite conductivity of the transmission line metal. The model uses

$$R(\omega) = \sqrt{R_{DC}^2 + R_{AC}^2} \quad \text{with} \quad R_{AC} = R_0 \sqrt{\frac{2j\omega}{\omega_0}}$$

to describe the general development. Again, R_{DC} and R_0 form the layer of abstraction for the physical implementation. The square root function models a smooth cross-over from the DC resistance dominated regime to the inductively dominated regimes. Finally, the dependence of the per-unit capacitance on frequency which is primarily due to the changes in dielectric permittivity with frequency are modelled as

$$C(\omega) = C_0 \left(\frac{j\omega}{\omega_0} \right)^{-\frac{2\theta_0}{\pi}} \tag{3.6}$$

where the loss tangent $\theta_0 \approx \tan(\theta_0) = \left(\frac{-\varepsilon_i}{\varepsilon_r} \right)$ specifies the relation between real and imaginary part of the dielectric permittivity of the material surrounding the center conductor at the given angular frequency ω_0. With these geometry independent functions in mind, the various performance regions and their limits can thus be defined. Depending on the geometry of an electric channel and the particular frequency band of operation there are the following performance regions of an electric transmission line:

3.2.1 Lumped Element Region

In this region, the length L of the transmission line is small enough so that the effects of distributed parasitic elements can safely be ignored. The transmission line can then be described as the equivalent of a Pi-Model in which all parasitic effects of the line are lumped into a single inductive, resistive and two capacitive elements (hence the name). This is the case when the magnitude of the exponent of the propagation function $H(\omega, l)$ remains lower than a constant Δ, typically set to a value of about 0.25 nepers (2.17 dB), that is

$$|l\gamma(\omega)| < \Delta$$

Since γ is a function monotonously increasing with ω, the inequality only needs to be checked for the transmission line of greatest extent at the maximum frequency within a given system. It can be shown that for an ideal, unloaded source (no source resistance, infinite load resistance), the resulting propagation function is approximately given by $H = \frac{1}{2+(l\gamma)}$.

In other words: as long as the transmission line is short enough to be considered "lumped", there is no spectrally distorting effect on the signal it carries. Its impact lies solely in the very slight attenuation of about 0.968 at the boundary of the region. This corresponds to a loss of roughly 0.3dB at the given length.

As can be seen in figure 3.3 a crossover from the lumped element region can occur either to the RC dominated or LC dominated domain, depending on operating frequency and length of the transmission line. Therefore, the maximum effective length a transmission

line that may safely be described as a lumped element can either be

$$l \approx \frac{\Delta}{\sqrt{\omega R_{DC} C}} \text{ when } R_{DC} > \omega L \text{ or } l \approx \frac{\Delta}{\omega \sqrt{LC}} \text{ when } R_{DC} < \omega L$$

Here, R_{DC} is the DC resistance of the transmission line (and its current return path) and Δ arbitrarily chosen just like above. Physically speaking, the first case then demands that the lumped cut-off angular frequency of the RC transmission line $f_{co} = \frac{1}{l^2 R_{DC} C}$ needs to be at least $\frac{1}{\Delta^2} = 16$ times higher than the maximum angular frequency ω of the signal to be transmitted. Similarly, the second case can be interpreted as the requirement of having an LC delay $l\gamma = l\sqrt{LC}$ 16 times smaller than $\frac{1}{\omega}$.

In this text, we are interested in how the attenuation (or equivalently put, the loss) of a transmission line changes with its most decisive feature: its length (and to some degree with its bisectional area as well). As the term *lumped*, however suggests, there is not much practical information to be gained here, as either the frequencies of operation are of only little interest to a multigigabit serializer or the extends of the transmission line itself are much too small. This will be much different for the regions to follow.

3.2.2 RC Region

The RC or *dispersive* region stretches from DC to (potentially) several MHz and requires that the extends of the transmission line are significant with respect to the wavelengths contained in the signals to be transmitted. The transmission line is then described as a series of small resistor-capacitor low pass elements as it is usually done in integrated circuit design for frequencies of several hundreds of MHz at larger technology nodes. Towards higher frequencies, this region ends as soon as the per-unit series inductance ωL becomes comparable in magnitude to the per-unit DC resistance of the transmission line, that is

$$\omega_{\mathrm{LC}} = \frac{R_{\mathrm{DC}}}{L} \tag{3.7}$$

Similar to the length boundary definition for the lumped element region, the length below which there is no significant distributed RC behavior can be derived to

$$l_{\mathrm{RC}} = \frac{\Delta}{R_{\mathrm{DC}}} \sqrt{\frac{L}{C}}$$

and is thus determined by the ratio between transmission line impedance and unit-length DC resistance. For long-haul channels in high quality cables and even for ordinary printed circuit boards traces shorter than one meter, the DC resistance is usually much smaller then the characteristic impedance (quite often $40 - 80\,\Omega$ making the RC region rather unimportant at these scopes. Due to the small bisection of on-package or on-die wiring

and especially with the advent of 3D through silicon via stacking, this region is however of noticeable importance to future short-haul multigigabit I/O standards.

In case of a low impedance source and a perfect termination $Z_L = Z_C$, the system transfer function G will read $G(\omega,l) = e^{-l\sqrt{j\omega RC}}$ and it can be seen that the logarithm of signal attenuation will scale with the inverse of the square root of frequency. The rather common case of termination with a purely resistive value chosen in accordance with the wave impedance $Z_L = Z_0$ of the line produces a much more involved equation which requires numerical processing and analysis.

3.2.3 LC Region

If the frequency of operation is increased beyond the point at which the inductive properties of the transmission line are less significant than its resistive properties as suggested by equation 3.7, the interconnection is said to be operated in the LC or *constant loss* region. Due to the physical dimensions of commonly used transmission lines within an electronic system (be it on die, within packages or on PCBs), this region with its constant loss versus frequency property is very narrow and often totally absorbed by the succeeding skin-effect region [24].

3.2.4 Skin-effect Region

As the frequency of operation is increased further, the eddy currents within the body of the conductors will limit the flow of charge carriers to a shallow region at the perimeter of the conductors. The so-called *skin-depth*, the thickness of the layer of metal at the surface of the conductor where the current will flow can be shown to adhere to

$$\delta = \sqrt{\frac{2}{\omega\mu\sigma}}$$

The effective series resistance seen at higher frequencies $R_{AC}(\omega)$ will therefore continuously increase with frequency. Ignoring more subtle effects (linear correction factors due to surface roughness and conductor proximity effects), the low frequency magnitude is

$$R_0 \propto \frac{1}{p\delta\sigma}$$

where again, p is the perimeter of the conductors in question. Quite obviously, the onset of this region depends greatly on the physical realization of the transmission line. Again, ignoring the subtle correction factors, an estimate is given by

$$\omega_\delta = \frac{2}{\mu\sigma}\left(\frac{p}{a}\right)^2 \tag{3.8}$$

where μ is the conductors magnetic permeability constant, σ its conductivity, p the perimeter of the conductor and a its cross sectional area. Since it can be shown that the propagation coefficient $\gamma(\omega)$ is proportional to the square root of frequency, the logarithmic signal attenuation itself, too, will exhibit a (inverse) square root dependency with frequency. For I/O standards operating in the regime of several hundreds to some thousands of MHz across packages and PCBs, the skin-effect region is the primary region of operation to be considered - as Millers analysis of the bit-rate capacity [36] also underlines.

3.2.5 Dielectric loss Region

As the electromagnetic wave progresses on the transmission line, it will inevitably interact with the surrounding dielectric material. This interaction can both take place on a molecular and atomic level depending on the angular frequency of the wave and stimulates a specific degree of freedom in the solid state body of the dielectric. Consequently, there is a transfer of energy from electromagnetic wave to the dielectric taking place. As frequency increases, so does the complex function of relative dielectric permittivity. The phase angle between its real and imaginary parts - the loss tangent, however, stays very constant over a broad range of frequencies. It can thus be shown that equation 3.6 produces a good estimation of the general process [1]. It can further be shown that the real part of the propagation coefficient which produces the attenuation component in the transfer function can be derived to

$$\mathrm{Re}\{\gamma(\omega)\} = \frac{1}{2}\frac{\theta_0\omega}{v_0}\left(\frac{\omega}{\omega_0}\right)^{-\theta_0/\pi}$$

from which it can be seen that dielectric loss increases signal attenuation on a logarithmic scale in proportion to frequency. Therefore, the cross-over range from skin-effect dominated to dielectric loss dominated region is very broad. Mathematically, the boundary can be specified by comparing the attenuation magnitudes caused by either effects and determining the frequency, where the contributions to overall signal attenuation are equal. This gives

$$\omega_\theta = \frac{1}{\omega_0}\left(\frac{v_0 R_0}{Z_0\theta_0}\right)^2$$

It must be noted though that due to the difference in proportionality to frequency between skin-effect and dielectric effect, already at a frequency ten times lower than ω_θ, the dielectric contribution to overall attenuation already is about one fourth! This aspect is especially important for the discussion of highly integrated transmission lines of very small

1 Modelling the complex relative permittivity correctly is quite involved and failure to enforce a stringent relation between its real and imaginary parts for all frequencies will produce transfer functions which can be nonreal or even noncausal.

dimensions. Naturally, their wave impedance needs to be rather low while their perimeter p is very small. This makes skin-effects look very dominant while pushing the value for ω_θ to fairly high frequencies (compare table 3.4). Due to the broad transition between the two performance regions, however, neglecting the dielectric effect will produce overly optimistic results.

3.2.6 Waveguide dispersion region

As soon as the electromagnetic wavelengths become comparable to conductor dimensions, electromagnetic field modes other than the regular TEM mode may also propagate on the transmission line. This linear superposition of modes can lead to phase distortion since the group delay of each mode may be very different. The onset frequency for non-TEM modes mostly depends on the physical geometry of the waveguide. For the cases of high integration and small cross-sectional dimension discussed here, however, this region can safely be ignored even for the years to come. Table 3.4 still lists values as a reference for some of the test channels used in this text.

3.2.7 Johnson Signal Model for a coaxial transmission line

The various different performance regions owe their existence to distinct physical effects gaining importance over others as frequency increases. Johnson proposed a set of equations which are suitable to describe a transmission line from DC up to the regime of 10 GHz with good accuracy [24] consistently covering all of the above mentioned regions except for the last. Although the model itself is not limited to a specific physical realization, the coaxial cable is chosen here due to its analytic tractability and comparability to what has been introduced by Miller et al. when defining the bit rate capacity.

In order to simplify the analysis in chapter 6, the free model parameters are constrained to a smaller set and the way in which (physical) parameters are fixed is explained below. The remaining free arguments of the model will be

- The target impedance Z_0
- The real part of the dielectric permittivity ε_r and
- Its loss tangent θ_0
- The frequency ω_0 at which the complex dielectric permittivity ε (and from it the frequency independent loss tangent θ_0), the magnetic permeability μ of the dielectric material and therefore Z_0 are defined. μ, however, is fixed to unity here.
- The length L of the channel
- The radius a of the coaxial cables inner conductor

The conductivity σ_{Cu} of center and shield (reference plane) conductor are constrained to the value of annealed copper at 20° C which is $5.87 \cdot 10^7 \frac{\mathrm{S}}{\mathrm{m}}$.

Note that one of the major reasons for the limited usefulness of the model towards very high frequencies originates from the rather simple dielectric model. At higher frequencies, the dielectric permittivity is a very much nonlinear function of frequency both in its real and imaginary parts and depends on the atomistic properties of the dielectric material itself. Especially the real part of the function can only be considered constant up to very low GHz regions where electromagnetic absorption processes by electric dipoles or molecular degrees of freedom can not be stimulated by the wavelengths of the propagating signal.

The geometry of the coaxial cable is shown on the left of figure 3.4. With the inner radius a given, we constrain the sheet conductor thickness d of the shielding conductor to the same value. For analysis of the dependence of system performance on cabling area A, the conductor bisectional area is calculated according to

$$A = \pi \cdot (b + 2d)^2$$

It can be shown that for a coaxial line, the per-unit capacitance and the per-unit inductance are

$$C_0 = \frac{2\pi\varepsilon\varepsilon_r}{ln(b/a)} \text{ and } L_0 = \frac{\mu_0\mu}{2\pi} ln(b/a)$$

respectively. With $Z_0 = \sqrt{\frac{L_0}{C_0}}$ given, the outer radius b of the cable is then constrained to

$$b = ae^{\frac{Z_0}{\chi}} \text{ with } \chi = \sqrt{\frac{\mu_0\mu}{4\pi^2\varepsilon_0\varepsilon}}$$

Apart from the prefactor of $\frac{1}{2\pi}$ due to the coaxial geometry and the material constants of the dielectric separating inner and outer conductor, χ looks quite like the definition of the free space wave impedance. It can directly be seen, why cables for high-density interconnections need to posses very low dielectric constant to limit the bisectional area. Also, high-impedance wiring beyond the standard 50 Ω which could reduce current consumption in transmitter output buffers and terminations, lead to an increase in bisection area and thus pose a challenge to high integration requirements. Additionally and as a consequence of higher area demand, high impedance transmission lines are also more susceptible to cross talk and generally electromagnetic coupling. For this reason and also due to increasing integration densities in PCBs, the trend of recent years has even been towards lower line impedances (such as 40 Ω in PCI- Express [14]). This, of course, is in notable contrast to what would be advisable from equation 3.2 for the bit rate capacity which increases linearily with Z_0.

The DC resistance of the coaxial transmission line is given by

$$R_{\mathrm{DC}} = \frac{1}{\sigma_{\mathrm{Cu}}}\left(\frac{1}{\pi a^2} + \frac{1}{2\pi d \cdot b}\right)$$

i.e. the inverse of the bisectional area of rugged center and a sheet shield conductor multiplied with the metals conductivity. For the low frequency value of AC Resistance R_0, we ignore surface roughness for the time being, setting $k_r = 1$. While this should be a good choice for coaxial cables, it is rather optimistic for traces on printed circuit boards - apart from the fact that feasible geometries on multilayer circuit boards are not all that well described with a coaxial approach. The proximity coefficient k_p (for details, refer to [24]) can be calculated to

$$k_p = \left(1 + \frac{a}{b}\right)$$

which thus gives

$$R_0 = k_p \cdot k_r \cdot \frac{1}{p\sigma\delta} = \left(1 + \frac{a}{b}\right)\frac{1}{2\pi a}\sqrt{\frac{\omega_0 \mu}{2\sigma_{\mathrm{Cu}}}}$$

To arrive at the values for the onset of non-TEM modes as given in table 3.2, we choose the earliest of these modes (TE_{11}, again compare [24]) which is calculated with

$$f_c = \frac{ck_c}{2\pi\sqrt{e_r}}$$

where the geometry dependent quantity k_c is the so-called *wave number* given by $k_c = \frac{2}{a+b}$ for a coaxial cable.

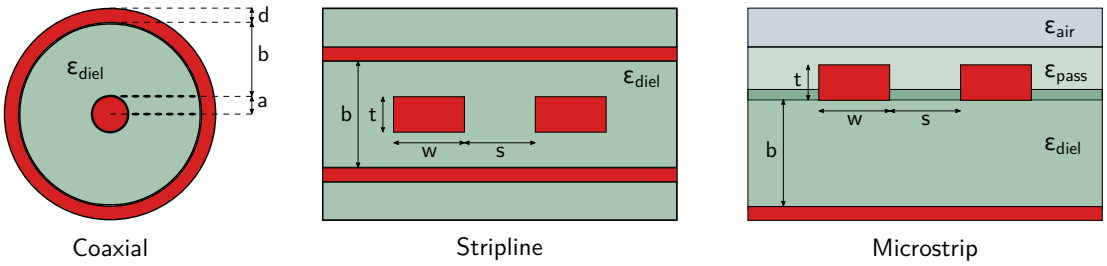

Figure 3.4: Common channel topologies as encountered in typical HPC systems, coax shown in single ended, stripline and microstrip in differential configuration

For the remainder of this text, there will be two different approaches to define and use

channel models. The analytic model presented here is used to derive scaling trends for serializer based links in close resemblence to what has been done in the context of bit rate capacity (chapter 6). For the actual verification and budgeting of serializer systems, however, we require less idealized channels. They need to be composed of various sections representing the different waveguide geometries encountered in a typical network link (compare figure 2.4 of subsection 2.2.2 and figure 3.4). These sections are

- The package breakout section - very narrow and thin traces contained in layered ceramic or organic material and covered by reference planes to either side. This is also called a *stripline geometry* and it is clear from figure 3.4 that the electromagnetic fieldlines surrounding the center conductor(s) will be vastly different from the coaxial case.
- The printed circuit board section - usually featuring much wider but not much thicker signal conductors and likely not to be entirely implemented as striplines. Due to potential AC coupling capacitors or connectors, there may have to be sections of microstrip lines with far less favorable wave guide properties, too. This is especially true due to the various dielectric constants of laminate, passivation and air that will be penetrated by the electromagnetic waves of the center conductor(s)
- The cable section - usually a coaxial cable of rather small bisectional area due to high integration demands
- Impedance discontinuities between all above mentioned sections which are due to package balls, component landing pads of packages, capacitors and connectors as well as due to vias for the interconnection between the various layers at which signal conductors reside.

Unlike coaxial geometries, stripline and microstrip transmission lines are far more involved to model. Additional effects such as surface roughness and the frequency dependence of the dielectric permittivity add to the complexity of more advanced models. Commercial computer aided design (CAD) vendors for microwave and RF software put considerable effort in these types of models. Also, in modern serializer design, all interconnects are usually routed in a differential, coupled topology to provide better EMI and crosstalk immunity in high density applications. This is indicated in figure 3.4 for microstrip and stripline and further adds to the overall complexity of the underlying equations. The close proximity of the two center conductors which carry the same signal with opposite polarity leads to an additional coupling capacitance between them and the resulting wave guide impedance will effectively be lowered. The details of differential signalling are beyond the scope of this text and it will be sufficient here to note the general derivability of so-called *mixed mode S-Parameters* from their bare single-ended counterparts.

It is for these reason that for serializer analysis and budgeting we choose to use model data generated with advanced CAD programs. The standard output of these modelling processes are the S-Parameter matrices as mentioned at the beginning of this section which can then be preprocessed and used within the simulation framework presented in the next chapter.

Since there would be a vast variety of parameters and implementation possibilities to arrive at very different transfer functions, the industry has agreed on certain standards to test transmission channels for compliance, that is: general suitability for a well-defined purpose. These standards are either given in the form of a frequency mask such as the dashed lines in the exemplary frequency transfer function plots below (figures 3.6 and 3.7) or - in the past and nowadays rather uncommon - in the form of pulse or impulse responses. The former is the case with the Optical Internetworking Forum (OIF) common electrical I/O (CEI) standard [46], an example for the latter is the 10G Ethernet standard [17]. Finally, a third alternative is to avoid any explicit channel compliance definition by demanding a channel to plainly allow operation at a specifc BER level with the transmitter and receiver electrical specifications of the standard. Such is the case for lower speed CEI channels and PCI-E 3.0 [14]. Compliance is then checked for the entire system against a so-called *eye mask*, usually a diamond shaped representation of the eye opening in terms of eye width and eye height at the defined BER level.

In the following, we will be working with four distinct channels depending on the scope of investigation. For lower frequency budgeting investigations, the 10G Ethernet reference channel (the specification refers to it as *stressor*) is used. It is defined in terms of its impulse response from which the frequency transfer function and the SBR can be computed. Both are shown in figure 3.5.

At higher frequencies, the CEI specifications for 25 Gbps long reach channels are used for orientation. The Johnson signal model is used to produce a worst case channel response for two different scenarios:

Both Johnson signal model channels have an impedance of $Z_0 = 50\,\Omega$ single-endedly and feature no coupling between the traces.

- The PCB type channel is shown in figure 3.6 and has a parameter set as listed in table 3.2. Due to its comparably high loss tangent and small center conductor radius, it is limited to a length of about 700 mm.

- The American Wire Gauge (AWG) type channel is shown in figure 3.7 and has a parameter set as listed in the second row of table 3.2. Due to the very low loss tangent and bigger center conductor radius, within the given constraint mask it can still span 2.5 m. The onset frequency of the dielectric region ω_θ is lower for the AWG channel even though the loss tangent θ_0 is smaller than for a PCB-type realization. This is because R_0 decreases at the same time even more drastically due to the larger

perimeter of the center conductor. It must, however, again be stressed here that ω_θ is only a measure of relative dominance between skin and dielectric loss magnitudes. As such, it is only an indicator of the relative importance each effect has for the data rate to be transmitted across this channel.

The well-known consequence of the numbers in table 3.4 thus is that dielectric effects are an increasing issue as data rates approach the tens of gigabits per second, even more so if the transmission channel (even partially) consists of printed circuit board traces. As metioned, this aspect is even further amplified by the fact that the coaxial line and its corresponding field distribution is a very optimistic approximation of realizable PCB channel structures.

Additionally, a commercial ECAD tool is used to construct a fully differential channel of $Z_{diff0} = 100\,\Omega$ with the above mentioned sections and intentional discontinuities between them where the values for the various sections of package and PCB substrate and the size of interconnect wiring between the two endpoints (compare figure 2.4) is chosen in accordance with commercially available, advanced manufacturing constraints. Figure 3.1 at the beginning of this section shows the according SBR and transfer function where the resonances of the discontinuities are clearly visible. Additionally, the TDR measurement of figure 3.2 highlights the various different sections of the communication channel and visualizes the discontinuities in time (and thus space) domain.

Channel	$f_0 = \omega_0/2\pi$	ε_r	θ	a	L
CEI25G-LR in PCB	1 GHz	3.5 (improved FR4)	0.007	$\sqrt{\frac{W_{typ}T_{typ}}{\pi}} = \sqrt{\frac{100\,\text{um}17\,\text{um}}{\pi}} = 23.3\,\text{um}$	0.686 m
CEI25G-LR in 32-AWG	1 GHz	2.0 (FPE dielectric)	0.0028	100 um	2.5 m

Table 3.2: Parameters of the Johnson Signal Model for the PCB-type and AWG-type coaxial cables used in this text

Channel Name	LC	Skin	Dielectric	TE_{11}	3dB cutoff	30dB cutoff
CEI25G-LR in PCB	5.71 MHz	26.92 MHz	24.8 GHz	380 GHz	430 MHz	16.6 GHz
CEI25G-LR in 32-AWG	430 kHz	1.36 MHz	17.14 GHz	158.75 GHz	480 MHz	16.5 GHz

Table 3.4: Approximate region boundaries with respect to frequency for the two Johnson signal model test channels presented here.

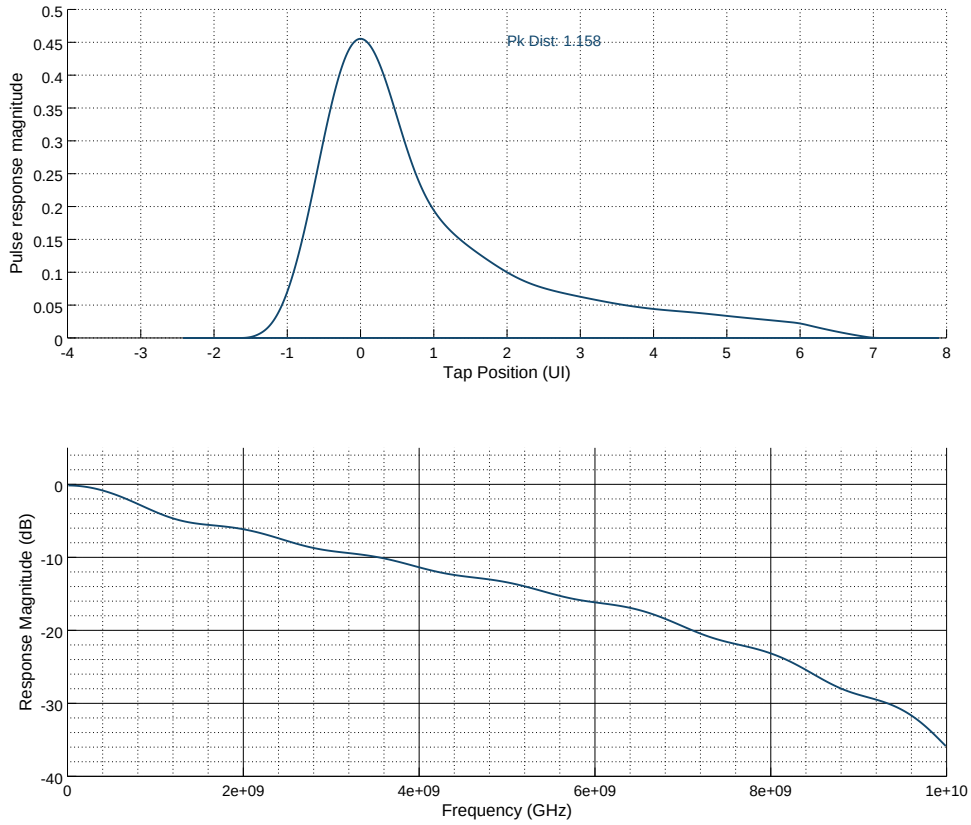

Figure 3.5: SBR at UI=100 ps and frequency response of a 10 Gigabit Ethernet reference channel

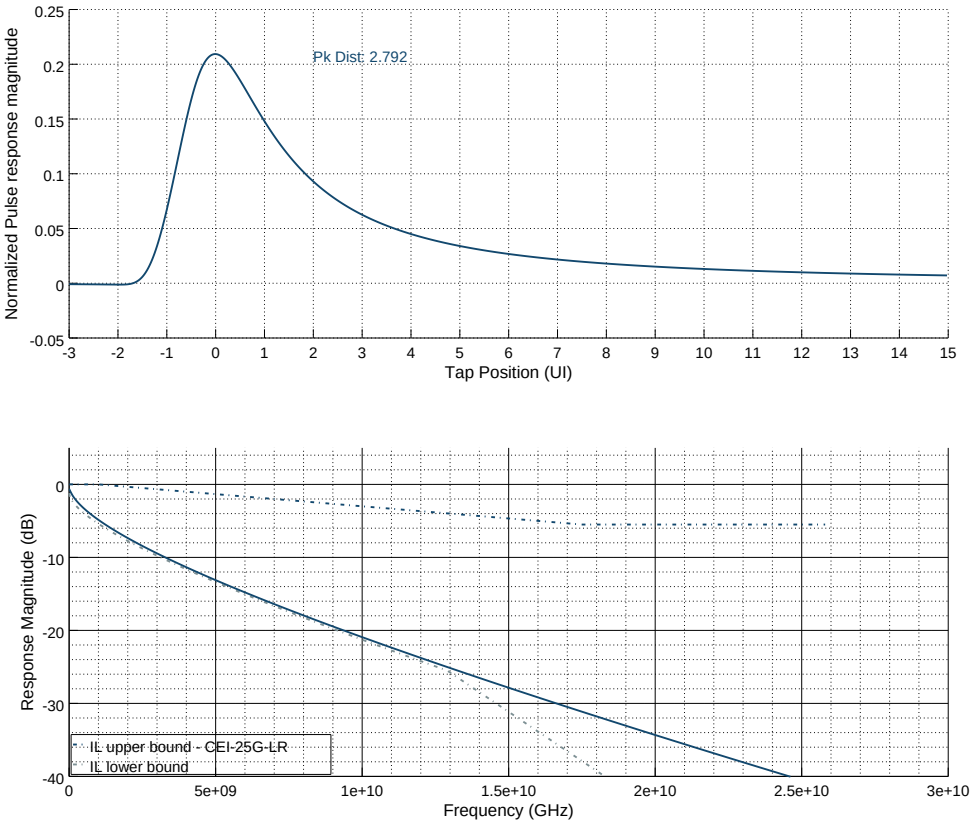

Figure 3.6: SBR at UI=40 ps and frequency response of a 25 Gigabit worst case CEI-25G-LR PCB channel (compliance mask shown in dashed lines)

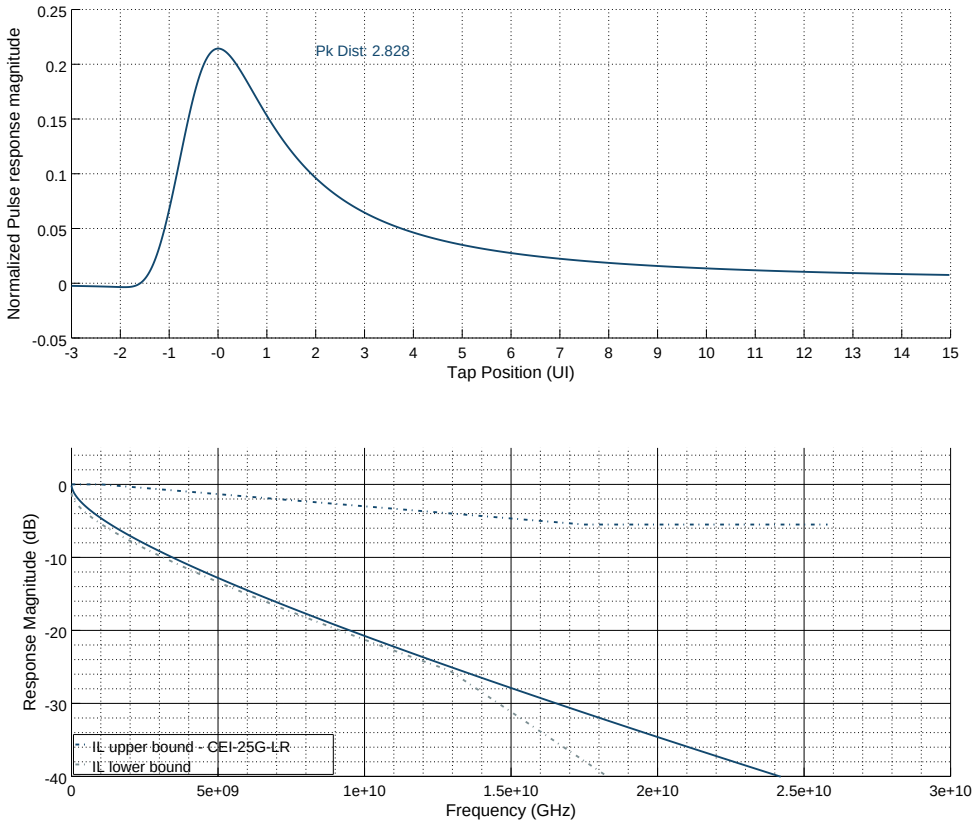

Figure 3.7: SBR at UI=$40\,\mathrm{ps}$ and frequency response of a 25 Gigabit worst case CEI-25G-LR AWG-type channel (compliance mask shown in dashed lines)

Without knowing much more details about the actual serializer system to be used, the Johnson signal model only preliminarily allows to derive some first estimations on how data rates may actually scale with transmission line properties. For this purpose we first note that the equalization efforts presented in the next subsection do not substantially elevate the signal level at the Nyquist frequency. For the moment, we therefore assume the receiver to be capable of still resolving the incoming data stream as long as channel attenuation does not fall below a given threshold level. In light of the publications in recent years, we set this limit in attenuation to $20\ \log_{10}|H(\omega)| = -30\,\mathrm{dB}$. The Johnson signal model for the coaxial line can then be used to compute the 30 dB cutoff frequency for a given set of parameters. For figures 3.8 to 3.10, the impedance Z_0, the length L and the bisectional area A (via inner conductor radius r_i) are tuned respectively, while the other parameters are fixed at $\omega_0 = 1\,\mathrm{GHz}$, $\varepsilon_r = 3.5$ and $\theta_0 = 0.007$ to reflect a more advanced but not too expensive PCB board material. The plot trace is colored red whenever the dielectric realm dominates over the skin-effect, it is colored grey whenever the skin-effect is more important in magnitude.

From this standpoint, the following relations for data rate scaling could be implied:

- The linear increase with A can especially be observed in the low bisectional area limit, thus within the skin-effect dominated region as predicted by Miller. When scaling the bisectional area A further, the increase in center conductor perimeter p decreases the importance of the skin-effect relative to the dielectric effect - a trend as already evident from the TDR measurement of figure 3.2 where the tiny package stripline traces showed severe skin-effect trends whereas the AWG cable produces a flat spatial impedance profile. A good fit to the model data (dashed lines in graphs) can be achieved with the function

$$f_c = \frac{B_0}{1 + A^{-\kappa}} \tag{3.9}$$

  where in this case $B_0 = 37.44\,\mathrm{GHz}$ and $c = 0.4$. We could interpret B_0 as some sort of upper bound to the bit rate capacity and κ as the geometry and parameter dependent scaling property as its value will certainly change when the relative importance of skin and dielectric effects are altered, i.e. via the loss tangent or the conductivity.

- When scaling Z_0, the coaxial line geometry has to be recomputed to maintain a constant bisectional area. The unit capacitance needs to decrease, therefore the inner radius will scale downwards as impedance increases. This in turn will for instance increase conductor resistance and skin-effect losses. Interesting enough, apart from not showing an aparent proportionality, the maximum of the observed lobe is in the vicinity of the well-established $50\,\Omega$ industry standard.

- The scaling behavior with L is better than an inverse proportionality to the squared

length as can be seen from the according logarithmic plot. From an HPC system designers perspective, this may actually be good news - always assuming that the initial presumption of a constant recoverable attenuation limit holds.

In order to more realistically estimate the potential performance given a specific serializer system, this chapter needs to be concluded with remarks on equalization techniques, the way in which equalization is chosen and the spectral content of NRZ data. With this full picture of a serializer system, chapters 4 and 5 will then describe how real numer modelling can help to analyze the various subcomponent contributions to overall performance less heuristically.

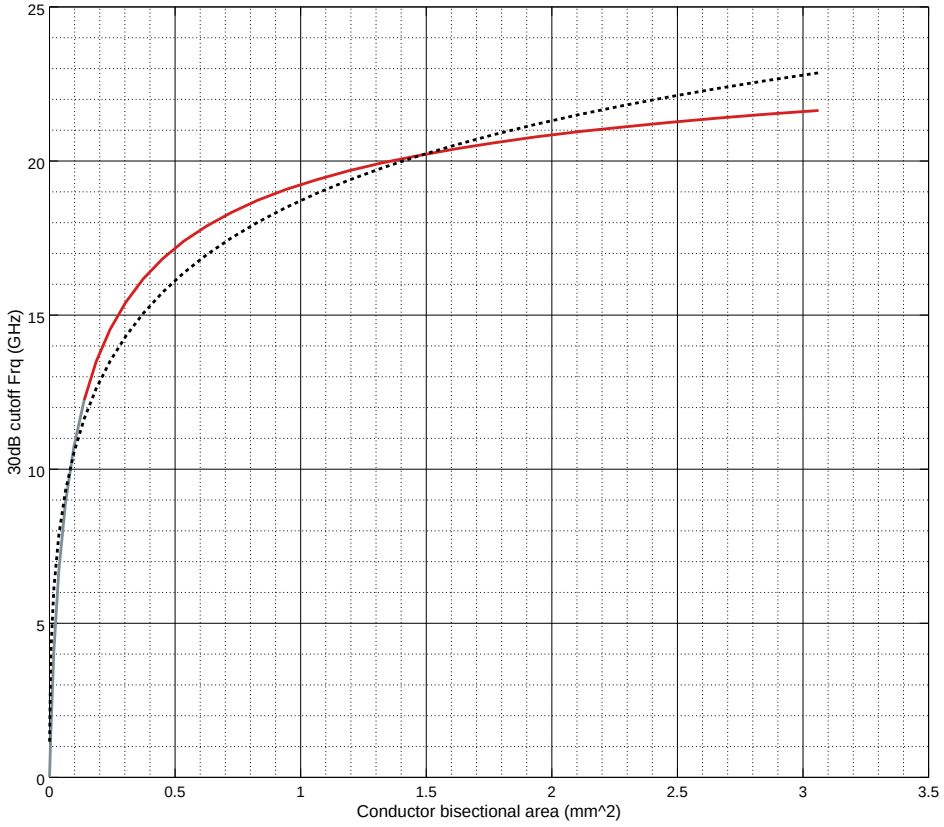

Figure 3.8: Functional dependence of the $30\,\mathrm{dB}$ cutoff frequency on bisectional area A of a coaxial transmission line with $Z_0 = 50\,\Omega$ and $L = 1\,\mathrm{m}$

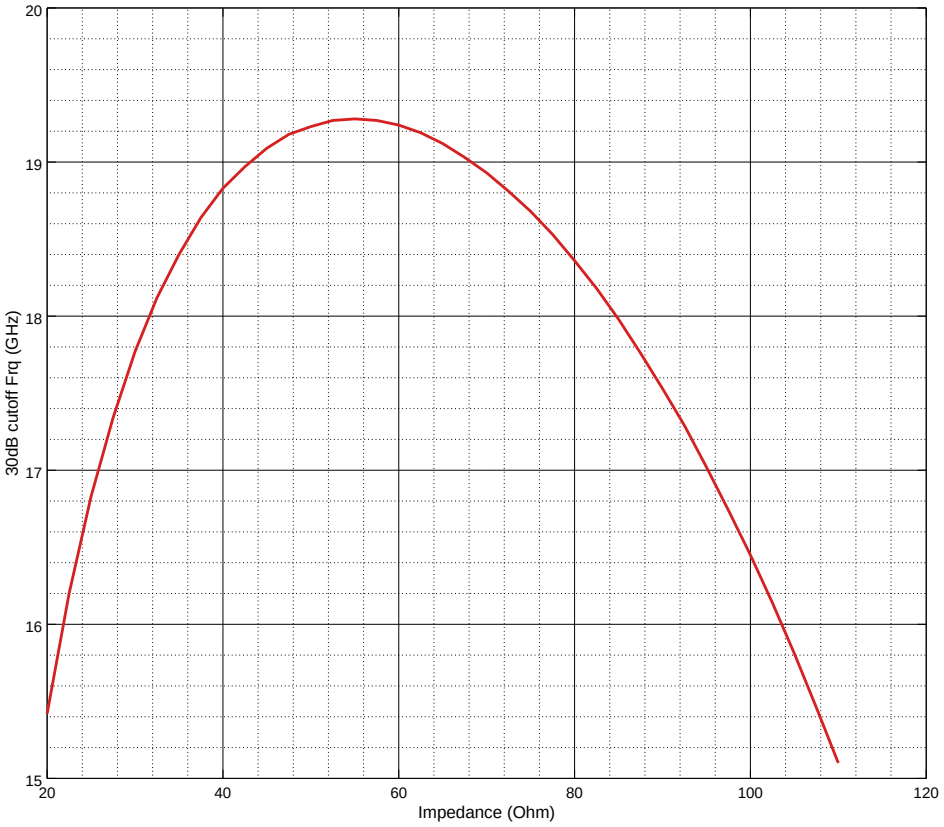

Figure 3.9: Functional dependence of the $30\,\mathrm{dB}$ cutoff frequency on wave guide impedance Z_0 of a coaxial transmission line with $A = 1\,\mathrm{mm}^2$ and $L = 1\,\mathrm{m}$

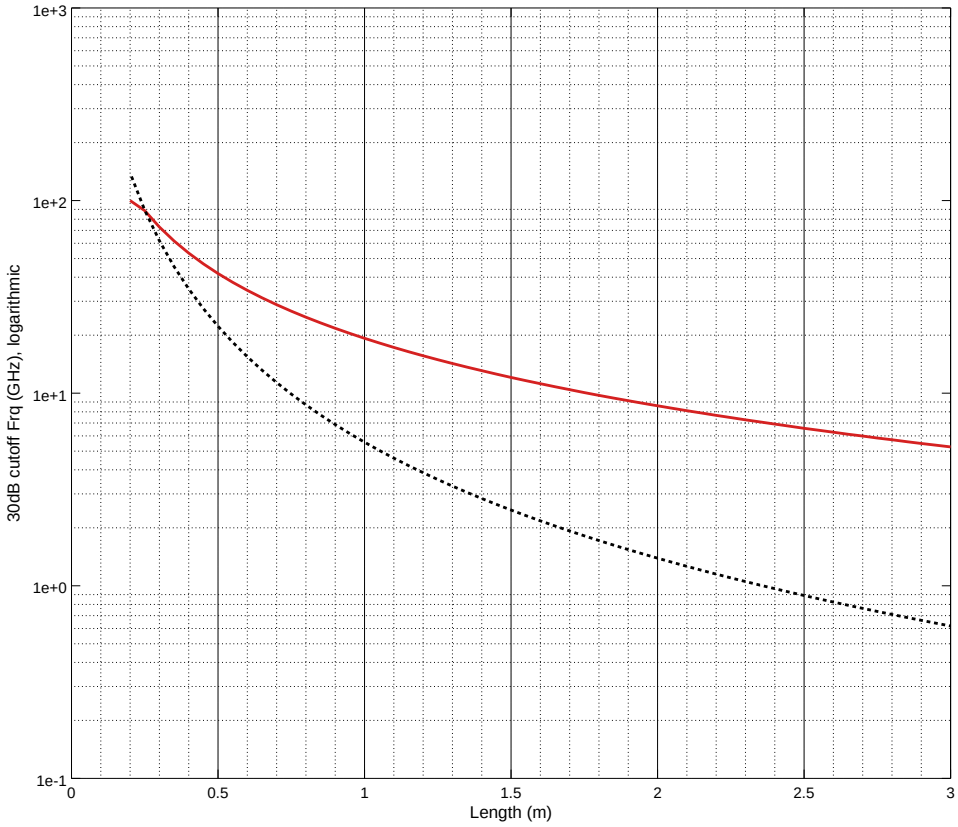

Figure 3.10: Functional dependence of the $30\,\mathrm{dB}$ cutoff frequency on length L of a coaxial transmission line with $Z_0 = 50\,\Omega$ and $A = 1\,\mathrm{mm}^2$

3.3 Equalization techniques

As can be seen from the previous section, most practical transmission line topologies, especially once they grow more complex and need to span a larger distance, do not allow multigigabit transmission without signal conditioning. This can either mean a predistortion of the signal in the transmitter, an analog filter in a receiver or even a digital feedback in the receiver to facilitate reception of succeeding bits. It is common basis of all these approaches to remove residual ISI from the overall SBR (the time domain view), or, equivalently put, to flatten the frequency response (the frequency domain view and ultimaltey where the name equalization stems from). Please note that the frequency responses, single bit responses and equalization results in this section already all originate from real number model simulations of the modelling and analysis framework and the serializer system presented in chapters 4 and 6.

3.3.1 Finite impulse response filter

Finite impulse response filters (in this context also called feed forward equalizers) are used in the transmitter of a serializer. This building block uses all digital inputs a_k and produces a preweighted, analog output

$$b_k = b(kT)p(t - kT)$$

with the pulse function $p(t)$ (see section 5.1) where

$$b(kT) = \boldsymbol{w} \cdot \boldsymbol{a}$$

with the so-called *tap weights* $\boldsymbol{w}$. The current, preceeding and succeeding bit(s) a_k form the vector $\boldsymbol{a}$ whose weighted sum is being produced at an internal or the external node of the transmitter. The particular realization of weighting and summation process is of course dependent on whether a CML or SST output or intermediate driver is used. For reasons of power supply rejection and the inevitable decrease in signal swing, however, virtually all designs in literature produce the weighted summation at their output nodes.

The concept of a four tap FIR filter is displayed in figure 3.11. This sequential shift structure with a single clock phase is called a full-rate architecture. It is however much more common to realize a multiphase design implementation strategy for the transmitter where the output data is produced by numerous clocks at equidistant phases. The probably most widely adopted concept uses two clock phases (*double data rate*) but with rising data rates, using more clock phases has become increasingly popular over the past years [15]. In this case, the FIR architecture has to be adapted accordingly. As far as the mathematical interpretation of the equalization process is concerned, however, the exact implementation

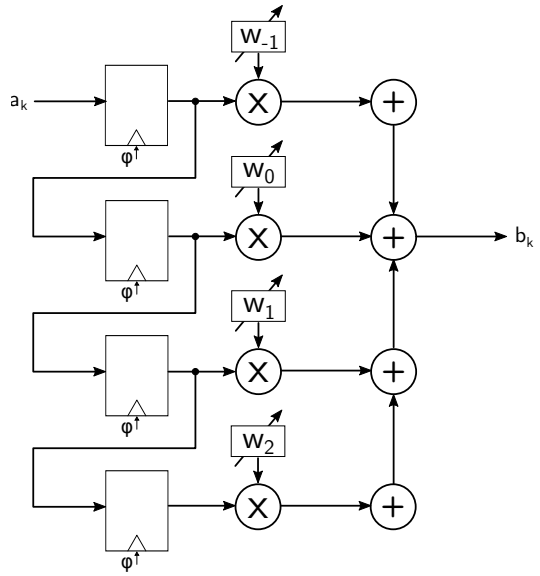

Figure 3.11: Conceptual view of feed forward equalization with an FIR filter

is of no importance here.

The FIR filter uses the current bit ($n = 1$) multiple preceeding and usually only a single succeeding bit to adjust the electric potential at the transmitter output. The weights w_k can be adjusted such that they optimally compensate the residual ISI of the channel SBR if the effect is not too strong. From a modelling point of view, the FIR filter is most easily described in Z-space as a delay chain with the associated weights.

$$H(z) = \sum_n w_n z^{-n} \tag{3.10}$$

The time discrete description of the FIR filter in z-space can, within certain limits, be translated to a time continuous description and therefore to a frequency transfer function via a bilinear transformation. This process is described in more detail and in different context in section 4.1. For an exemplary channel in compliance to the CEI-28-MR standard, the FIR can be shown to both remove the ISI from the SBR and to produce a much flatter overall frequency response (see figures 3.12 and 3.13). As can be seen especially from the resulting frequency transfer function, the process of equalization at the transmitter will equalize the signal at the expense of decreasing the overall signal to noise ratio at the transmitter output already. Also, due to the power constraint of the transmitter

$$\sum_n w_n = 1 \quad \text{while necessarily} \quad w_0 > w_n \forall n$$

the magnitude of correctable intersymbol interference is strictly limited.

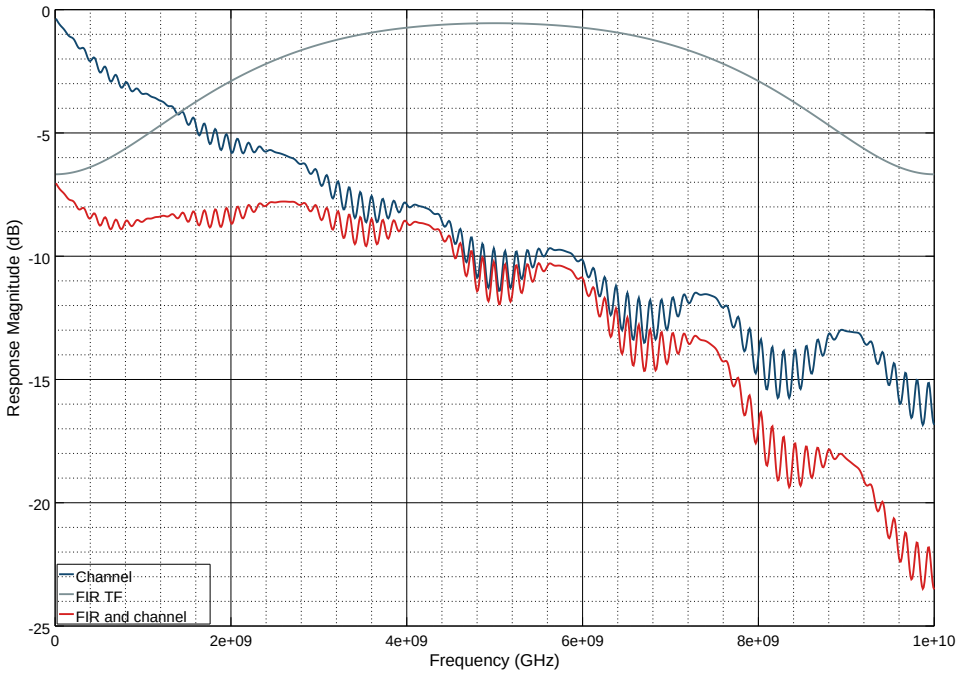

Figure 3.12: Bode plot of an FIR frequency response along with the channel transfer function (CEI28-MR) and the resulting transfer function

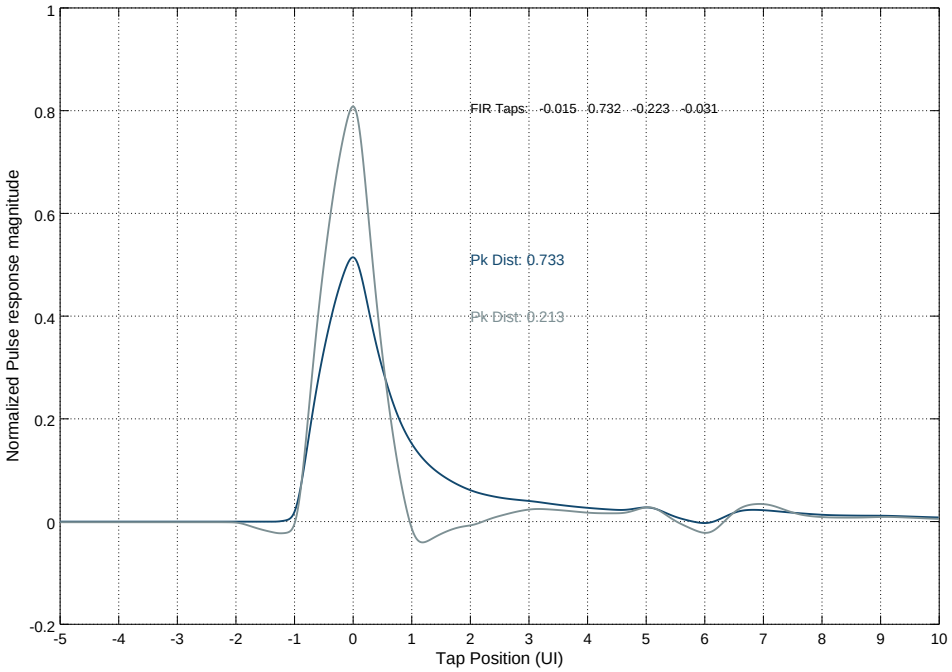

Figure 3.13: Corresponding single bit (pulse) responses of uncorrected and composite channel with a bittime T of $100\,\mathrm{ps}$

3.3.2 Continuous time linear equalizer

The continuous time linear equalizers (CTLEs) are a means of signal preconditioning prior to analog to digital conversion in the receiver. Thus, these type of circuits are a part of the amplification chain of the receiver analog frontend. They can be implemented in either active or passive ways. Since usually channel signal attenuation is sufficiently high when equalization is required, the active structures such as the one presented in figure 3.14 are more common as they offer additional gain and may thus reduce the number of amplifier stages and therefore noise in the signal path leading to the sampler inputs. As each gain stage in the analog receiver frontend also amplifies the noise present at its inputs, there can of course be no improvement in the signal to noise ratio as a result of equalization. The overall gain requirement and the resulting receiver sensitivity thus need to be carefully analyzed. This analysis is with respect to residual internal offsets, sampler intrinsic noise and residual ISI.

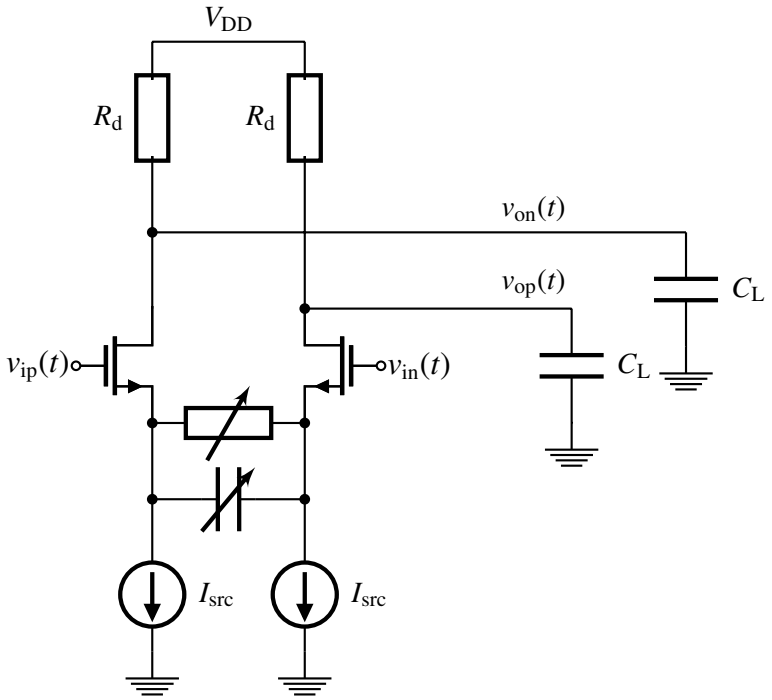

Figure 3.14: A fully differential continuous time linear equalizer with resistive and capacitive degeneration.

The general transfer function of an active CTLE can be derived from its small signal model. Ignoring the gate-drain capacitance of the input transistors C_{gd} and their finite output impedance r_{ds} for the moment, the transfer function of the CTLE circuit shown in

figure 3.14 can be calculated to [9]

$$H(s) = \frac{(g_m R_D)(1 + sR_s C_s)}{(1 + sR_s C_s + g_m R_s/2)(1 + sR_D C_L)}$$

Here R_S and C_S are the (usually adjustable) source degeneration resistance and capacitance while g_m is the small signal gain of the input transistors. The locations of the zero and the two poles of the transfer function above calculate to

$$\omega_z = \frac{1}{R_s C_s}$$

$$\omega_{p1} = \frac{1 + g_m R_s/2}{R_s C_s}$$

and

$$\omega_{p2} = \frac{1}{R_D C_D}$$

respectively. Figure 3.15 shows an example transfer function for a CTLE with a bandwidth of 6.8 GHz (the location of the second pole), a zero located at roughly 490 MHz and the first pole located at 6.54 GHz.

In order to compensate a first order low pass channel with a cutoff frequency of $f_{\mathrm{BW,s}}$ for a signal at a data rate of R (and thus $T = 1/R$) it can be shown that the required peak or DC gain can be calculated [9] from

$$\frac{A_{\mathrm{pk}}}{A_{\mathrm{DC}}} = 0.5 \cdot \frac{f_{\mathrm{BW,pk}}}{f_{\mathrm{BW,ch}}} \tag{3.11}$$

where $f_{\mathrm{BW,pk}} \approx \frac{2}{3T}$ has to coincide with both $f_{\mathrm{p1,p2}}$ and f_{z} has to be chosen to coincide with $f_{\mathrm{BW,ch}}$. From the transfer function, the required gain levels can then be calculated according to

$$A_{\mathrm{DC}} = \frac{g_m R_D}{1 + \frac{g_m R_s}{2}}$$

and therefore

$$A_{\mathrm{pk}} = A_{\mathrm{DC}} \frac{\omega_{\mathrm{p1,p2}}}{\omega_{\mathrm{z}}}$$

Apparently, there is a power tradeoff to two sides. The peak gain level will directly increase the power of the amplification stage itself. On the other hand, if we choose a lower peak gain level, the gain in the low frequency limit (towards DC) will have to be significantly lower to satisfy equation 3.11. However, the lower the attenuation is chosen,

the better the signal to noise ratio of the entire system will have to be so that a given BER target can still be realized - after all we are deliberately lowering the signal to noise ratio at lower frequencies. An improved overall SNR can for instance be achieved by increasing internal capacitances, especially within the receiver samplers (see section 4.5) or the amplification chain which then of course increases the overall power consumption once again. In essence, there is no easy analytical solution to the problem and a mostly automated design framework would be very helpful in designing and dimensioning these type of preamplification stages so that a given channel can be supported with the smallest power penalty possible. Also, when CTLEs are used in conjunction with other equalization techniques, the possibility of tradeoffs between these various stages can only be analyzed in the context of the entire system.

3.3.3 Decision feedback equalization

The decision feedback equalization stage is located in the receiver and its particular implementation depends on the number of clock phases of the architecture just as is the case with a transmitters FIR filter. Figure 3.17 depicts a full rate direct DFE. As the number of sampling clock phases in the receiver increases, so does the number of required samplers and summation nodes. Although there are multiple summation nodes in the figure, architecturally these are usually collapsed into a single summation stage which of course needs to take its back action on an entirely analog signal. The design space of summation and weighting concepts is fairly large and stretches from classic, resistive amplifier and current summation concepts to more involved integration or switched capacitor realizations [40, 4, 48, 11]. From a higher level point of view, however, and ignoring challenging aspects of signal to noise ratio in analog preamplification and shaping stages for the moment, all concepts lead to the same functional description of the building block itself.The incoming analog signal is corrected by a weighted sum of the previously received bits.

One of the central challenges in DFE design is the sampler (depicted by the threshold symbol in the figure). In the full rate design shown here, the output bit resolved by the sampler needs to be weighted and fed back to the summation stage where the incoming analog signal must be corrected sufficiently early so that the sampler again has enough time to resolve the next bit - thus unimpaired by ISI of the first postcursor. Therefore, this signal loop from sampler output to input needs to fit into a single bit time. Digitally speaking, we would be writing

$$t_{\mathrm{cq}} + t_{\mathrm{sum+w}} + t_{\mathrm{su}} < T$$

with the usual bit time T, the "clock to output delay" of the sampler t_{cq}, the "setup time" of the sampler t_{su} as well as the propagation delay of wires, summation and weighting buffer $t_{\mathrm{sum+w}}$. However, it must be noted that clock to output delay and setup or hold time are

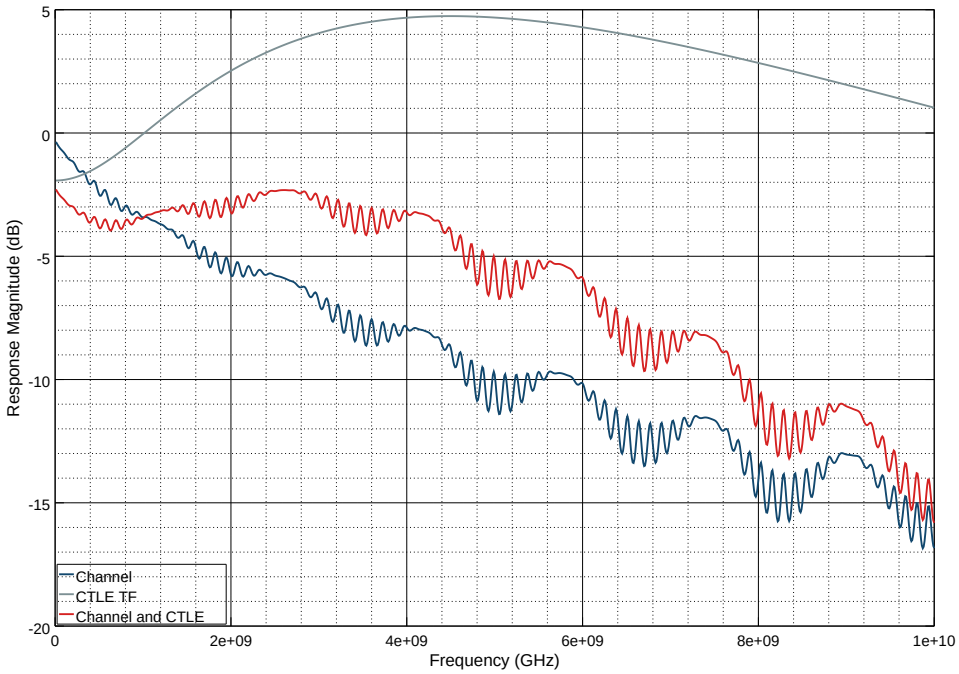

Figure 3.15: Bode plot of a CTLE frequency response for an active circuit such as the one shown in figure 3.14. While the DC gain is primarily determined by the resistive degeneration, the location of the zero depends both on degeneration and overall small signal gain.

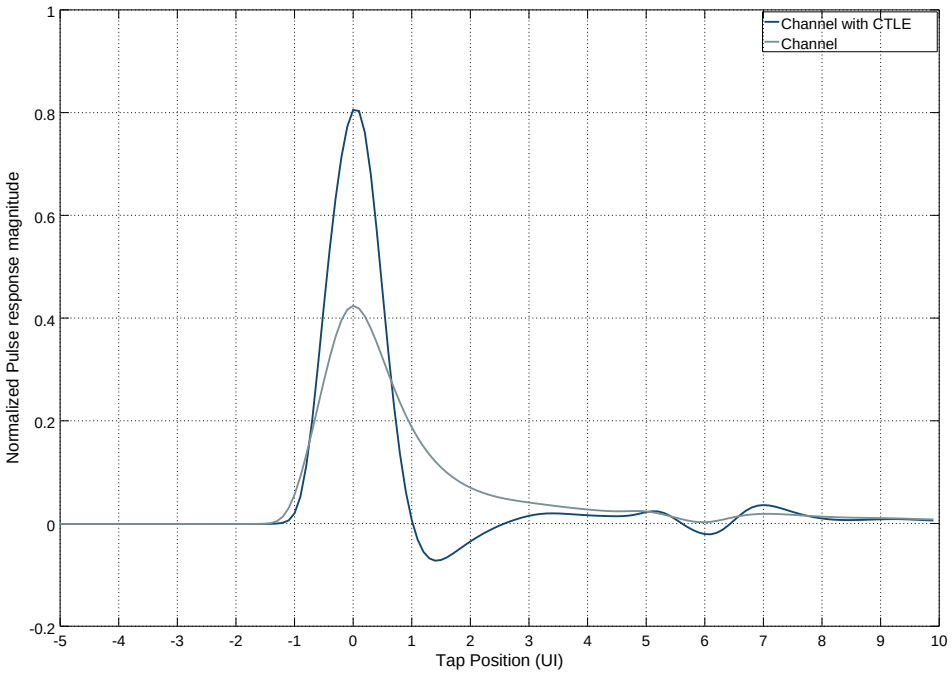

Figure 3.16: Corresponding pulse response of the system channel with CTLE. The unit bittime T is $100\,\mathrm{ns}$

concepts of the digital, standard cell driven domain where there usually are well defined logic voltage levels. This of course is not the case for a sampler operating on possibly very small analog inputs. Roughly speaking, the small input signals will lead to rather long *resolution times* (ergo a large t_{cq}). Also, without proper modelling of the sampler, it is unknown and completely signal dependent, which magnitude is to be attributed to t_{su}, i.e. how long the signal needs to be "stable" to be correctly resolved. It is for these reasons that full rate direct DFE designs, even in modern technology nodes, are very hard to realize. Also, this is one of the primary reasons for presenting a coherent modelling approach for high speed samplers in this text (see section 4.5).
To circumvent this problem, most modern receiver architectures employ a mechanism called *speculation*. Instead of having a single sampler and a direct feedback to the summation node at its input, there are two samplers per clock phase and at least two clock phases. The samplers have static, opposing offsets, each favoring the reception of one logic level over the other. Once the decision of one set of samplers is known, it is forwarded to the other set of samplers effectively doubling the decision interval of a sampler. This can be seen in figure 6.4 for the quarter rate receiver architecture analyzed in chapter 6.

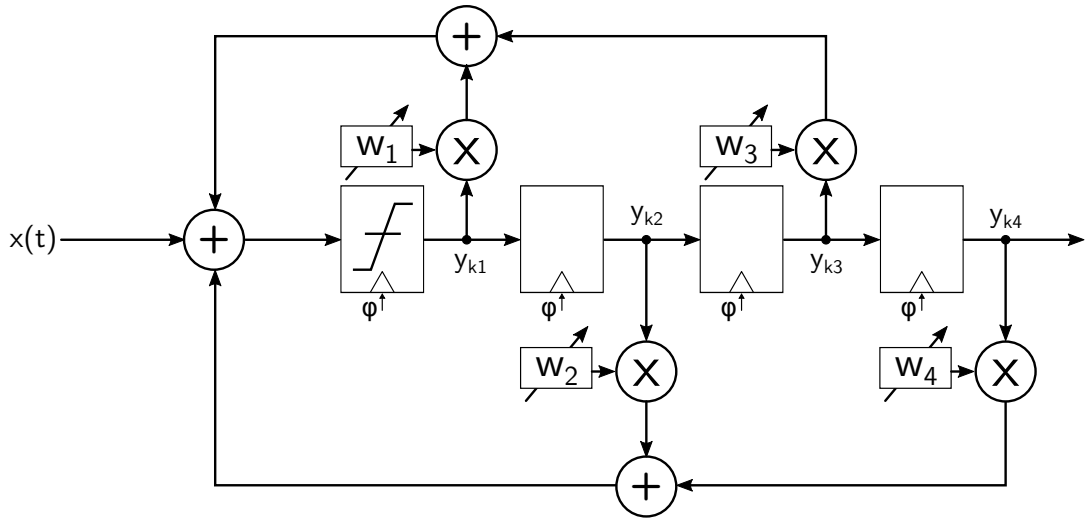

Figure 3.17: Conceptual view of a full rate direct feedback equalization filter

For the exemplary CEI-28-MR compliant channel also used in the previous examples, the DFE can be shown to remove the ISI from the postcursors of the SBR (see figure 3.18). Additionally, since the amplification process in the summation node is very nonlinear, the added noise to the analog input signal is limited to that of the summation architecture itself. Since this is mostly independent of the particular choice of $\boldsymbol{w}$, the DFE is the only equalization stage which does not increase the overall signal to noise ratio in proportion to

equalization strength. Note that the corrected SBR shown in figure 3.18 actually does not exist as a directly observable voltage anywhere in the serializer system. A virtual multipoint probe which pieces together this virtual SBR has to be implemented in accordance with the particular, implemented architecture - a task formidably suited for SystemVerilog.

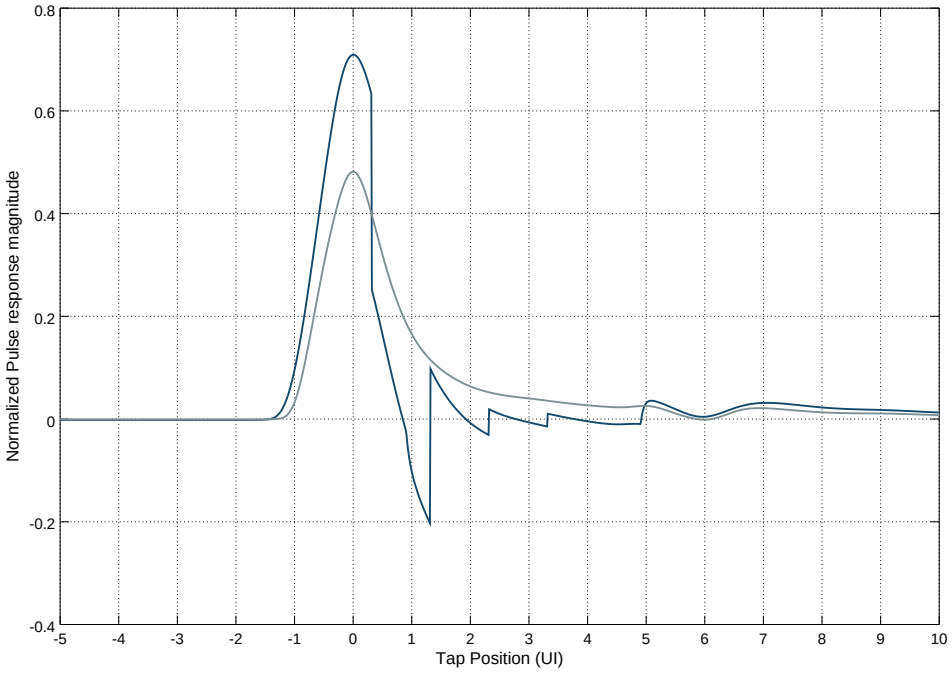

Figure 3.18: Single bit pulse responses of the DFE corrected and uncorrected system channel. The unit bittime T is $100\,\mathrm{ns}$.

3.3.4 Equalizer adaption

There have been many suggestions in the past on how to best achieve a good level of equalization. The apparent similarities between DFE and FIR often mirror in proposed algorithms which work for both equalization schemes at the same time. The major difference in the proposals lies in the metric according to which the optimization process of the equalizer taps takes place. This metric can be of different nature: perhaps the most common adaption target is the eye height (or equivalently, the resiudal ISI) at the sampling instant [5] by using a sign-sign least mean square (LMS) adaption. There have also been suggestions for jitter reducing optimization schemes whose target then is the actual eye width [5]. Both eye width and height can be targeted with an algorithm which relies on the estimation of medium range bit error rates of an entire eye (somewhere between 10^{-4} and 10^{-8} depending on desired overall convergence speed and accuracy requirements) [66].

The benefit of the sign-sign algorithm lies in its simplicity and compatibility with the so-called *eye monitor* - a debugging mechanism which is routinely implemented in a serializer. An eye monitor is essentially an independent sampler whose relative sampling point and voltage threshold can both be adjusted - in the context of LMS adaption, it is called the error sampler. By recording a stream of bits and comparing them with the recovered bits of the data path, the bit error *ratio* at a particular point in the receivers eye (hence the name) can be captured provided that reception of data is mostly free of errors.

The simplified sign-sign LMS algorithm is identical to a zero forcing solution in the sense that both minimize the ISI at the particular sampling point [5]. This can be expressed in terms of the so-called *peak distortion* value

$$D_\mathrm{p} = \frac{1}{|q_d|} \sum_{n \neq d}^{N} |q_n|$$

which puts the residual ISI components at the various pre- and postcursor tap locations into relation to the main tap (assumed to be that of largest magnitude). This number is annotated for most SBR plots in this text.

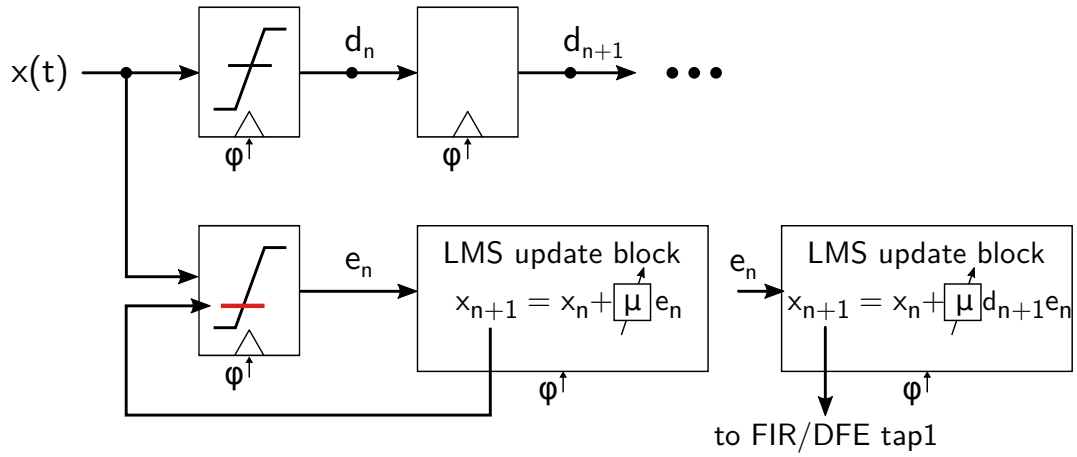

Figure 3.19: Concept of a full rate sign sign LMS based equalization adaption loop for error level and first post cursor

Figure 3.19 depicts a simplified, full rate sign-sign LMS equalization adaption loop which seeks to zero out ISI at the first post tap. With the sampling location fixed to that of the regular data path, the control loop simultaneously adapts the error level of the error sampler and the tap strength itself. It reaches equilibrium once the mean output of the error sampler is zero (if the logic levels are denoted as ±1). An important consequence of this dependence on the statistical mean of the digital output is that the transmission alphabet itself which is used to train the equalization loop, must not statistically favor one digital

level over another. In this sense, pseudo random bit sequence (PRBS) patterns are suitable for link equalization training procedures. The tap and error level update procedure adheres to the sign-sign LMS equation

$$x_{\mathrm{n}+1} = x_{\mathrm{n}} + \mu e_n d_k$$

Here, x_{n} is the tap value (or rather its real valued equivalent) at time instant n. e is the sign of the error sampler output, d that of the received data bit. μ is the loop gain factor and has to be chosen such that the control loop is stable while exhbiting a decent speed of convergence. Also, for proper convergence k has to be chosen in equivalence to its actual feedback properties on the signal, i.e., if we seek to optimize the first post cursor of the TX FIR, $k = n + 1$.

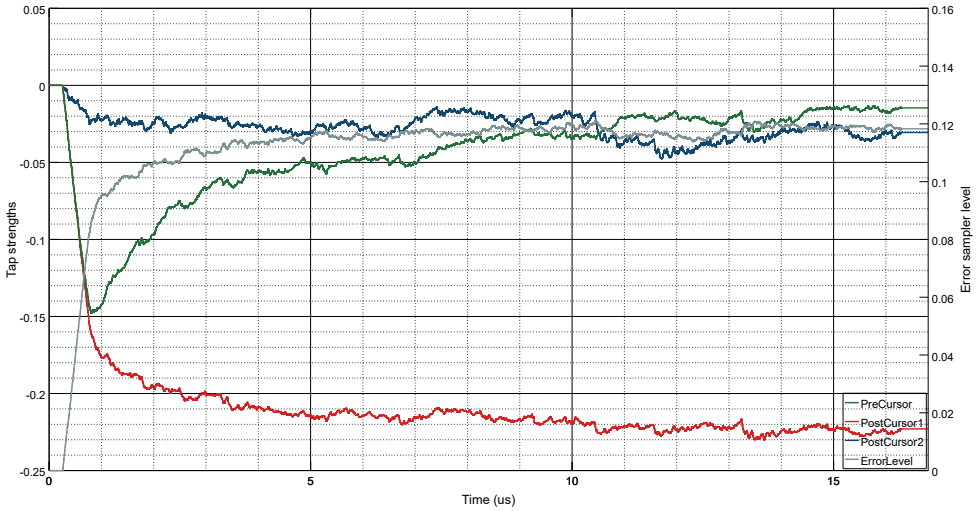

Figure 3.20: Convergence of FIR tap coefficients and sampler error level for the equalization effort shown in figure 3.13

Figure 3.20 depicts the convergence process of a set of TX FIR equalizer taps for the exemplary equalization setting given in figure 3.13. The CTLE equalization mechanism itself does not directly lend itself very well to this kind of adaption scheme. Only a small amount of literature can be found on digitally assisted CTLE adaption schemes [39]. Some papers propose a straight out linear search along the potential tuning vectors of the CTLE [66] with the more common case being a somewhat analog, power equivalence based adaption loop. in principal, the framework proposed here could be used to devise a procedure with which the CTLE, too, could be integrated into the simple, hardware efficient sign-sign LMS adaption scheme in the future.

3.4 Line Coding and power spectral density

The budgeting approach suggested in chapter 5 requires some consideration with respect to the spectral content of the signals we intend to transmit. Serializer link operation is usually analyzed, debugged and specified in accordance to a pseudo random bit sequence of well defined length. In addition, the analysis here restricts itself to the case of binary signalling: For a random sequence of NRZ coded bits, there is a straight forward derivation of the resulting signal power spectral density [42, 55]. Its starting point is the autocorrelation function $R_{xx}(t)$ of the stream of rectangular bit pulses $x(t)$. For a truly random, infinitely long stream of bits, it can be seen from equation 2.4 that $R_{xx}(t)$ needs to reach its maximum for a displacement $\tau = 0$ while it must be zero as soon as the displacement and the bit time adhere to the relation $\tau > \pm T$, i.e. the absolute displacement is greater than one unit interval. Since we are dealing with idealized rectangular pulses here, the degradation in $R_{xx}(t)$ from maximum to minimum must be linear. As mentioned in section 2.1 the Fourier transform of the ACF is the PSD which in the case of the resulting triangular function of width $2T$ and height a is known to be the sinc function

$$\mathscr{F}\left\{R_{\mathrm{xx,triangle}}\right\} = aT\left(\frac{\sin(T\omega/2)}{T\omega/2}\right)^2 \tag{3.12}$$

which is plotted alongside its autocorrelation function in figure 3.21. The still unspecified amplitude parameter a can be shown to have the value $a = T$ in case the pulse amplitudes are set to unity.

For the more practical case of PRBS with finite length L, the PSD can be derived from these previous thoughts [55]. The sequence repeats every L bit times. In time space, this is equivalent to a convolution of the PRBS sequence with a dirac comb $\sum_n \delta(t - nL)$ whose Fourier transform is also a Dirac comb $(1/L)\sum_n \delta(f - n/L)$. The greater the length of the PRBS sequence, the more tightly packed the Dirac comb in frequency space will be. This intuitively makes sense as with random bit sequences of growing length, the number of sine and cosine base functions needs to increase in count and the Fourier coefficients need to part from zero for ever lower frequencies as well. Since in time space the new ACF is found by convolution, its PSD equivalent is produced by multiplying the constituent Fourier transforms. The envelope of equation 3.12 stays the same with its continuous character dissolved in to a comb of weighted Dirac peaks spaced at a spectral distance of Δf_δ. The PSD of a random NRZ sequence of finite length L and a bit unit interval of T thus is

$$|A(j\omega)|^2 = \frac{T^2}{L}\left(\frac{\sin(T\omega/2)}{T\omega/2}\right)^2 \tag{3.13}$$

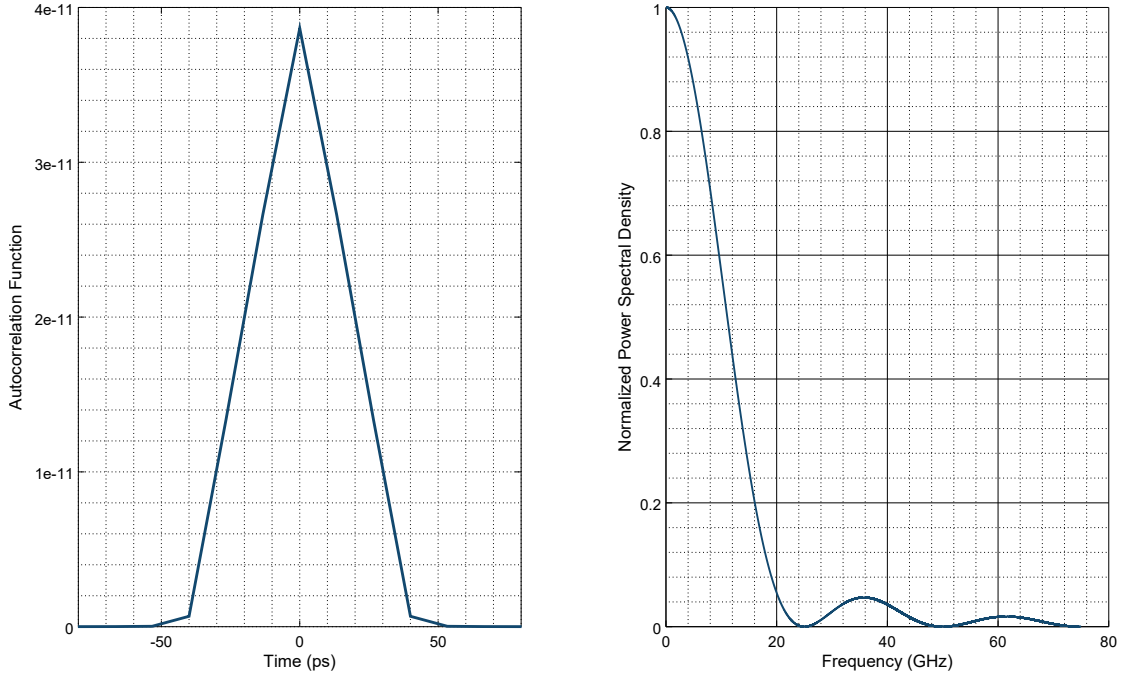

Figure 3.21: Autocorrelation function $R_{xx}(t)$ and Power Spectral Density $S_{xx}(f)$ of an infinite stream of random bits in NRZ coding sent at $25\,\mathrm{Gbit/s}$

while the Dirac peaks are spaced at a distance of

$$\Delta f_\delta = \frac{1}{L \cdot T} \tag{3.14}$$

This is plotted for the case of a PRBS-7 pattern transmitted at 25 Gbit/s in figure 3.22. Two important conclusions can be drawn from these derivations: From a channel signal integrity perspective, it is not a decisive difference how long a PRBS sequence actually is once it is beyond a certain length L. Common PRBS sequences are PRBS-7, 15, 23 and 31 with sequence lengths $L = 2^N - 1$ for PRBS-N. For a unit bit time of 40 ps at a 25 Gbit/s transmission rate, the dirac comb spacing will therefore be 196.75 MHz, 750 kHz, 3 kHz and 12 Hz respectively with the lowest frequency content at the same value. Probing serializers with long random sequences is to a much lesser degree a test for signal integrity (i.e the frequency response of the channel) than it is a test for resilience of other serializer components with respect to long run lengths of zeros and ones. Good examples are the locking behavior of the clock data recovery (CDR) which requires a reasonable transition density to acquire and maintain a good sampling position or the DC wander caused by AC coupling capacitor charging due to the temporary DC imbalances introduced by long sequences of constant logic level on the transmission line. The second interesting consequence from a design point of view is the required bandwidth of the receiver analog frontend and a perspective on signal to noise ratio additionally to the one provided by the

eye diagram perspective (see section 2.2.1). Normalizing the PSD to unity and integrating over frequency provides the plot of figure 3.23. It shows how much of the total signal power P is contained from DC up to a given frequency. Normalizing the x-axis to $T \cdot f$, one can see that about 80 % of the total signal power is already contained at half the signals Nyquist frequency. In order to capture about 93 % of the signal energy, an integration up to about 3/4th of the Nyquist frequency has to be made while for 99 %, already the Nyquist frequency has to be exceeded by a factor of 1.6. While the extension of the frequency bandwidth only leads to a very modest increase in additional signal power, the constituent components of the required amplification chain (transistors, resistors) produce all white Gaussian noise of unlimited spectral extent [1]. An extension in bandwidth by a factor of N will therefore also increase the total in-band noise by approximately the same factor. Evidently, there will be a particular point in frequency at which an additional increase in bandwidth will inevitably lead to a reduction in signal to noise ratio. However, there are other, often bandlimited sources of noise in the analog frontend of the receiver such as the power supply induced noise which is filtered by the on-die capacitances. The required bandwidth can therefore often only be found approximatively and it has become common practice from an energy efficiency point of view to limit the bandwidth to about two thirds of the Nyquist rate.

For narrowband systems, the so-called spot signal to noise ratio, the relation of signal power to noise power at and in close vicinity to a certain frequency is the decisive quantity. In theses type of applications, information is modulated onto a carrier of specific frequency. These modulations can be either in amplitude, phase or both but usually are strongly limited to a certain frequency band around the central carrier where transmission channel and noise properties of all components usually change mildly. In broadband applications, however, depending on the line coding the signal-to-noise ratio is more difficult to analyze. Of course, it is always possible to record or derive an eye diagram at the output of any given subcomponent in the system. However, for some design choices, the view described above delivers valuable insight into how subcomponents may be reasonably constrained.

There have also been publications on more advanced line coding schemes for electrical multi-gigabit serializers. These include either the use of duobinary coding [32] or the even more involved PAM-4 signalling scheme [32] [62] [13]. The common benefit of both is a compression of the PSD main lobes signal power to only half the frequency bandwidth. This comes at the expense of introducing more signalling levels and thus requires the transmitter to produce a broader range of output voltage levels (something very akin to an FIR based output) and the receiver to resolve these distinct levels back to a digital representation. In the receiver this necessitates an increased number of samplers and the additional signalling levels decrease the signal to noise ratio from the start. It

1 Unlimited at least with respect to the frequency ranges described here

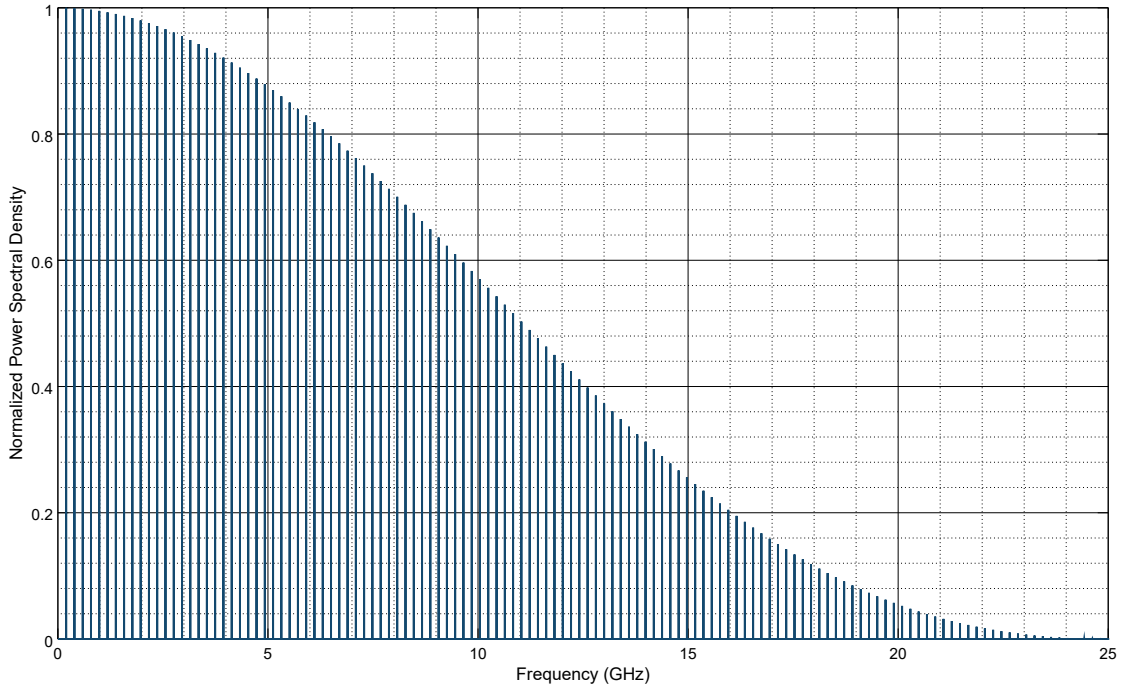

Figure 3.22: Power Spectral Density $S_{xx}(f)$ of a PRBS-7 stream of bits in NRZ coding sent at $25\,\mathrm{Gbit/s}$

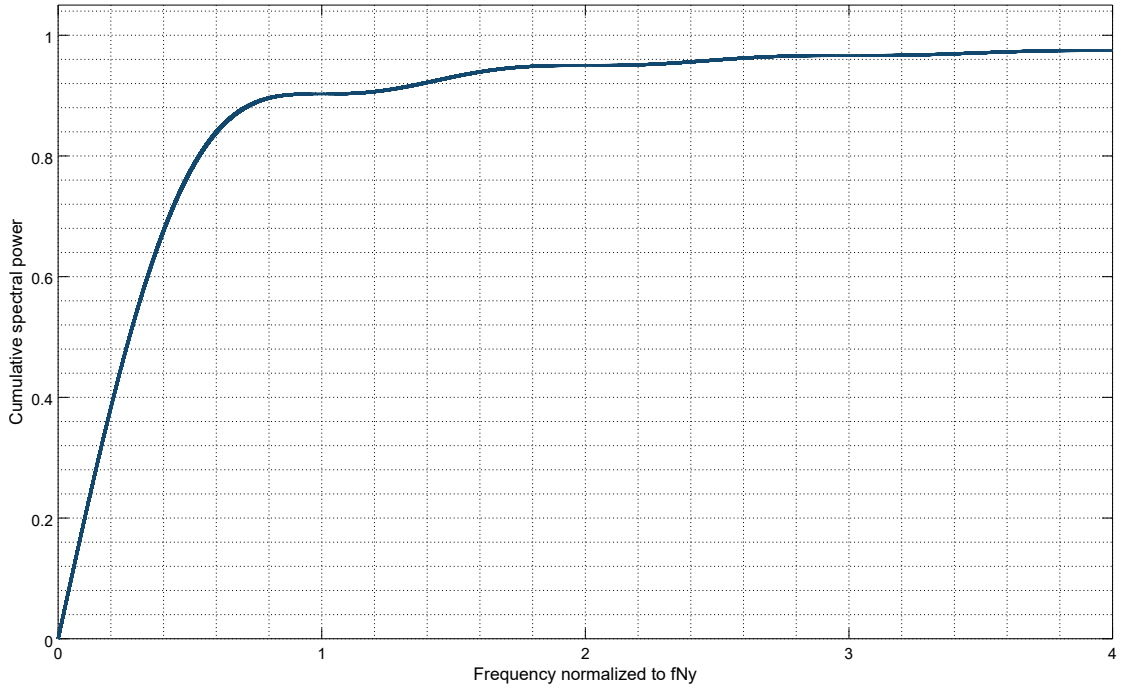

Figure 3.23: Cumulative power contained in an NRZ PSD normalized to the Nyquist frequency of the bit stream

is therefore highly dependent on data rate and the channel used, whether or not a more complicated signalling scheme provides an advantage with respect to reaching lower BER

levels. As more recent papers suggest [28] in the near future, it may therefore be a viable option for transmitter and receivers to support more than a single line coding and rely on sophisticated analysis schemes to determine which provides the best performance within a given environment.

4 Serializer system component modelling

This chapter will give a short introduction on the openMGT real number modelling framework for serializer systems and an extensive overview on its numeric backend extension using the open source Matlab clone Octave for advanced modelling features and post simulation analysis processes. It will be shown how the combined benefits of SystemVerilog real number modelling supplemented by the flexibility and computational efficiency of Octave can be used to model critical components. These include the transmission channel, the amplification and equalization stages in the receiver as well as the highly nonlinear receiver sampler circuits. The resulting models will speed up the simulation process by a factor of 1000 compared to their schematic based counterparts. At the same time, unified testbenches for both model and schematic implementation will show no significant deviation of the extracted performance metrics between the two views. These models will allow to investigate the individual influence of a broad range of subcomponent metrics or equalization choices on overall serializer system performance.

The previous chapter has highlighted the importance of a thorough understanding of serializer system sucomponent interaction and how their respective metrics may impact design choices. The driving top-level metric of a cumulative bit error rate has been shown to consider random noise effects within the components next to the various, strictly deterministic sources of error. Additionally, it was pointed out how vastly different the time constants of the subcomponents are, ranging from a few MHZ for PLL/CDR bandwidths over tens of MHz for equalization and offset cancellation calibration loops to hundreds of MHz for power distribution network induced noise all the way up to the range of GHz for channel cutoff frequencies and intrinsic noise source bandwidths. Also, a good BER estimate cannot be obtained without properly considering the extents of and tradeoffs between the various equalization schemes and their different impact on signal to noise ratio. Thus, a multitude of fairly lengthy simulations have to be carried out which, due to the inevitable fast transients, will have to use fine grained time steps. This, of course, is far from being a perfect fit for SPICE based simulations.

As also previously pointed out, without fast simulation models interactions of subcomponent metrics with higher level performance targets may be difficult to capture in transient simulation which makes proper design space analysis at the higher layers an increasingly difficult task. Most often, high model speedup is a result of choosing a different level of abstraction. This level of abstraction needs to be chosen carefully and potentially even

numerous times for various models of the same component to arrive at the best solution for the given analysis scope. A more abstract, cumulative scope such as the CDR jitter rejection requires bare simulation performance and may thus rely on linearized models which may in turn be derived from their more detailed counterparts. For smaller, more simplistic scopes, such as the frequency response of an amplifier, the runtime of an AC simulation alone for instance, does not justify an additional component model. Its transient response or its digital tuneability may, however, benefit from an abstraction of its transistor representation since the real number based model view can be used to introduce a new abstraction layer directly into the general design flow. Next to a speedup, this also ensures consistency between the different representations and modeling approaches of all the various subcomponents (see again section 4.2).

For performance reasons, the additional modeling layers should only mimic the most important properties of the physical component or, as mentioned, feature multiple levels of abstraction depending on the higher level simulation task. An obvious example for this is the line coding with its spectral impact on data transmission and thus intersymbol interference which directly benefits from a more abstract modelling approach of channel and amplifier chain by real number models (see below). A less obvious analysis question which can be answered more effectively with the right model abstraction is the impact of data coding on system functionality. The maximum run length of subsequent bits in the data code may disturb the equilibrium state of the CDR (i.e. the recovered sampling point) since a CDR tracks transitions in the data stream.

There is a wealth of tools available to tackle different modelling and simulation tasks during serializer design apart from schematic based design entry and all-electrical simulation (SPICE, SPECTRE and many more). These include statistical simulation tools such as SeaSim and StatEye or microwave engineering tools such as ADS for electrical channel analysis and characterisation for instance. These tools require a substantial amount of input parameters such as the transmitter voltage swings, rise and fall times, channel geometry as well as the equalization properties of the receiver or models extracted from already conceived designs (such as IBIS models). Also, a lot of ECAD tools for integrated circuit design provide Matlab integration to allow more complex, numerical modelling and even *cosimulation* where parts of the design are true schematic level while others are signal flow models (such as those defined by Simulink). This can used for a high level of abstraction and mathematically advanced simulation (testbench) output checks.

The major problem, however, is to keep all the subcomponent definitions and model layers coherent with the implementation itself. A good example for this is the linear phase model of a CDR (see section 2.2.6) which by itself requires a substantial set of parameters to allow computation of its loop bandwidth. Knowing the bandwidth alone may already be good enough to generate an approximate statistical eye with random jitter contributions as it may be recovered by the given receiver design. Creating the CDR subcomponents

such that they actually adhere to the very same constraints requires keeping all definitions and testbenches across different programs consistent. This, of course, is a major source of error.

There is a well established flow for defining testbenches and creating designs in the digital hardware semicustom design flow. This flow, however, profits from a complete physical abstraction by using so-called *standard cell libraries* and the large signal to noise requirements that are enforced by these (logic cell) definitions. The abstraction requirements for analog and mixed signal systems are, however, much less generic. Also, modelling can take place at various different levels of abstraction:

The goal of VerilogA for instance is to allow a much more mathematical definition of the relation between currents and voltages between two ports of a component - a powerful mechanism that easily allows to define completely discontinuous models for which convergence in DC or transient simulations may not even be achieved. VerilogA models run in the context of the analog simulator itself and thus produce results continuous in time and value [1].

In extension to this, VerilogA/MS was conceived which brought together the time and value discrete nature of Verilog HDL with the analog view of VerilogA. The language extension for instance allows to create components with both digital and electrical ports and defines translation procedures from one domain to the other. These include filter and transition functions to avoid grave discontinuities in analog signals as well as threshold and cross functions which are capable of creating digital events. The simulation process is basically split into two distinct parts, the *analog kernel* as mentioned above and the *digital kernel* with its event driven, time slot based simulation approach. The physical quantities of voltage and current with interaction between the two (i.e. impedance) can still be represented while the computationally efficient time discretized view can be used wherever the digital realm is required.

Finally, there is a further step towards stronger computational simplification in VerilogA/MS and SystemVerilog - a grammatically more powerful successor to Verilog/A/MS. It is called *real number modelling*. The time discrete nature of Verilog is ammended by a net type of continuous real number values - the wreal net and its register-like counterpart of real (double) values. This mechanism allows the value and time continuous description as needed and produced by analog environments to be translated to a value continuous but time discrete representation. It therefore replaces the time consuming approach of variable step size and iterative convergence processes in full analog simulation with a *signal flow* view as used in numeric simulators such as Matlabs SimuLink extension. There are also so-called *user defined nettypes* which allow to bundle numerous values into a user defined

1 There are of couse the usual minimal real value discretization of computer systems and the unavoidable but dynamic simulation steps of every analog simulator

structure and assign a net type to them. A net-coercion mechanism which resolves whether or not two ports of potentially different type can be interfaced with a *connection module* facilitates the use of this approach. Moreover, there are *resolution functions* which allow to define custom mechanisms of resolving a multi-driver, multi-receiver situation on a given interconnect of the design hierarchy. The combination of these mechanisms thus allows to mathematically mimic complicated analog interdependencies on a time discrete basis without having to resort to explicitly solving Kirchhoff equations of a larger network. For more detailed information, the reader is referred to the SystemVerilog documentation [1].

For every subcomponent in the serializer system, the challenge lies in finding an appropriate level of abstraction that exploits both the continuous character of the real number domain and the discretized character in the time domain to realize a model of high performance and reasonable accuracy. Subsection 4.1 will give an overview on how this can be done in the context of SystemVerilog with signal flow models for almost generic analog filters and describes the limitations of this process. SystemVerilogs original purpose was to improve and automate simulation capabilities and provide stimuli of greater variability to digital designs under test. Its long term goal is to establish similar verification procedures in the mixed signal and analog domain. The benefit of keeping the modelling procedure close to the implementation by using the same language for testing, modelling (and in the digital context even describing) a component lies in the consistency not only between model and implementation but also between the various modelling views of a subcomponent. Model views then can have various degrees of abstraction depending on the high level metric to be analyzed. If feasible, the development of the model and the testbench are done concurrently. The testbench is designed such that it can be used for extracting the relevant subdesign metrics. The common SystemVerilog based formulation (and some extensions of VerilogAMS and the simulators) makes the testbenches reusable for real number, electrical and possibly purely functional models alike. Therefore, the testbench and the real number model are verified against one another. The testbench checks a design under test with an explicitly given parameter. The model needs to use this parameter to produce the expected reactions to the stimuli. The testbench then has to retrieve the given metric magnitude with the specified test procedure. In this way, the model is shown to serve both as a means of representing the implementation during design exploration and as a basis for component verification after design and before tapeout. The framework presented in section 4.2 relies on these ideas to model and implement a parameterizable serializer with a top-down, digital first methodology. This was conceived in collaboration with another dissertation. The so-called *openMGT framework* is briefly described here for orientation. For more details the reader is referred to [38].

Some models of a serializer system, most notably the transmission channel, require more computational power than almost all other elements in the design combined. In case of channel and sampler modelling as presented in sections 4.4 and 4.5, convolution

requires multiplication of long real number vectors. This is not supported efficiently in SystemVerilog and the problem is magnified by the (current) absence of multiprocessing capability in contemporary SystemVerilog simulators. Furthermore, a strict separation of random and deterministic effects due to simulation time optimization and problem complexity is one of the key design principles of the openMGT framework. Interoperation with statistical budgeting algorithms and postprocessing of data as discussed in chapter 5 requires SystemVerilog to be extended by a numeric engine. MATLAB offers such capabilites and, as mentioned above, commercial CAD tools offer API bridges to these applications. However, these are often implemented with interprocess communication calls and are thus enforcing a context switch within the CPU. For frequent operations such as the convolution of a signal with a channel impulse response in a transient simulation, the communication overhead plays a vital role and needs to be diminished in importance compared to the computation effort of convolution itself.

Fortunately, the SystemVerilog standard defines a direct programming interface (DPI) extensions which allows to extend the simulator binary with custom C code. This will pave the way of extending SystemVerilog with the powerful open source MATLAB clone *Octave*. It leverages the openMGT C modelling extension (OCM) discussed in section 4.3 which will be used to model channel and sampler instances efficiently with SIMD instructions and threaded execution. It is also the gateway to the more complicated, statistical post processing which will be required by lane budgeting and peak distortion analysis as described in chapter 5 and also provides a consistent view on composite, linearized models such as the PLL or CDR.

4.1 General Real Number Modelling Considerations

While simulator time steps in SPICE may vary in size, they are fixed with RNM simulations. SPICE offers the option of fixing the step size but this comes at a severe performance impact. The convergence algorithm for solving the Kirchhoff equations has to be run for every time step of an analog simulation. When there only seems to be little change in voltages or currents in the vicinity of a particular node, the simulator chooses wider time steps if they are not constrained by the user.

The simulation speedup of real number modeling primarily lies in the abstraction of electrical components with a signal flow view. Thus, the time consuming process of solving the Kirchhoff equations for an entire, potentially huge matrix with nonlinear elements using a successive approximation technique and variable time steps can be circumvented. In its stead, the models have a fixed time step which needs to be chosen once for a particular subcomponent or even the design itself (the *simulation time step*). Also, a single net (connection between any two ports) has only one non-interacting quantity associated with it (a real number or a logic level) to which multiple drivers (outputs) may make contributions

(via the resolution functions mentioned above) and multiple receivers (inputs) may sense the value. Thus, there is no concept of impedance, wire load or wave reflection initially conveyed in this picture. On the other hand, newer versions of SystemVerilog offer the possibility of creating *user defined net types* which allow to associate a tuple of real number and logic values with a net. A resolution function may then be used to define, in which way multiple drivers interact on a given value. The most basic example would be a cumulative sum of all output drivers to a net for instance. With its help, various current sources summing up on a single node can easily be modeled. The associated voltage drop increase across an attached output resistance however cannot be described without extensions to either the resolution function of the connection or the resistor model itself.

Since the RNM approach lends itself best to signal flow descriptions of systems, the effects of a linear, time invariant system on a signal are best described in the frequency domain (s- space) and converted to z-space for time discretization. Then, the resulting FIR or IIR filter can be described with a few lines of code. The conversion from s- to z-space requires a transformation which in this case is chosen to be the *bilinear transformation*. It is defined as

$$s = \frac{2}{T} \cdot \frac{z-1}{z+1} \tag{4.1}$$

with its inverse being

$$z = \frac{1 + Ts/2}{1 - Ts/2} \tag{4.2}$$

Here, $T = \frac{1}{f_s}$ is the sampling rate of the time discrete system. The bilinear transformation basically maps all elements of the s-plane $s = \sigma + j\Omega$ to unit circles in the z-plane $z = Ae^{j\omega}$. Since σ in s-space is used to enforce convergence of the transformation integrals even for unstable systems, we can safely set $\sigma = 0$ as the subcomponents described here are all either passive or strongly bandwidth limited systems. In the z-plane, everything is then mapped to the unit circle $z = e^{j\omega}$.

From equation 4.2 we thus have

$$j\Omega = s = \frac{2}{T}\frac{e^{j\omega}-1}{e^{j\omega}+1} = \frac{2}{T}\frac{2e^{-j\omega/2}\,j\sin(\omega/2)}{2e^{-j\omega/2}\,j\cos(\omega/2)} = \frac{2j}{T}\tan\left(\frac{\omega}{2}\right) \tag{4.3}$$

Digital simulators in principle allow us to choose the sampling rate and simulation time step freely (or at least down to 1 fs which is far smaller than required by the radio frequency and microwave regions described here). Increasing the time step, however, naturally comes at the expense of increased simulation time and will therefore be especially hurtful for the purpose of simulating systems with a large spread in their characteristic time constants. Equation 4.3 can therefore be used to calculate the minimum time resolution

required to model a system up to a particular frequency without the nonlinearity necessarily introduced by the bilinear transformation process. The question which needs to be answered beforehand is how to select the frequency range of interest for a particular subcomponent or even the entire system. A distinct and viable choice here would be the 3 dB-cutoff frequency of the system (see section 3.4) as frequencies beyond this point provide very limited contribution to the overall signal seen at the receiver samplers - at least if the channels indeed have high loss characteristics as is the case in backplane and long-haul applications.

Assuming a target angular frequency of Ω_0, further simplification and rearrangement of equation 4.3 gives

$$\frac{\Omega_0 T}{2} = \tan\left(\frac{\omega_0}{2}\right)$$

The approximation of $\tan(\delta) \approx \delta$ holds to within 0.34 % error for $\delta \leq 0.1$ and to within 10 % for just about $\delta \leq 0.52$. Thus for a stringent linear mapping between the two spaces, the general requirement

$$T = \frac{1}{10 f_0} \tag{4.4}$$

should be fulfilled and for a more relaxed mapping the requirement

$$T = \frac{1}{2 f_0} \tag{4.5}$$

is sufficient. For a system whose most wideband component features a cutoff frequency of 20 GHz for instance, the simulator time steps and sampling rate of the time discrete filter representing the continuous transfer function would need to be set to 5 ps for the stringent case or 25 ps for a more relaxed setting.

Conversely, if there is a time-discrete component in the system and we are interested in its continuous frequency response, the frequency mapping process is mediated by the $\operatorname{atan}(\delta)$ function whose linearity is even slightly better (10 % for just about $\delta \leq 0.6$). This makes the mapping process reasonably linear up to frequencies of about

$$f_0 = \frac{1}{1.66 T} \tag{4.6}$$

The time continuous frequency response estimation of an FIR filter with a data rate of 10 Gbps (i.e. $T = 100$ ps), would therefore be only accurate up to 6 GHz which is slightly above half the Nyquist frequency of the given system.

For modelling analog system subcomponents with an linear time invariant (LTI) response

in signal flow view, an adjustable and versatile approach to translate the response of a rather generic time continuous filter is required. Since for the majority of purposes in a serializer system, the number of zeros and poles up to twice the Nyquist rate is limited to only a few, the transfer function

$$H(s) = A_0 \frac{\left(1 + \frac{s}{\omega_{z0}}\right)\left(1 + \frac{s}{\omega_{z1}}\right)}{\left(1 + \frac{s}{\omega_{p0}}\right)\left(1 + \frac{s}{\omega_{p1}}\right)\left(1 + \frac{s}{\omega_{p2}}\right)} \tag{4.7}$$

in canonical form is chosen as a starting point. Here, the set of $\omega_{z,n}$ describe the zeros in the system while the set of $\omega_{p,n}$ describes the poles. A_0 is the DC gain of the filter. In order to acquire a closed form solution for the z-space filter the bilinear transformation (equation 4.2) can directly be used to replace all occurences of s in equation 4.7. Given the appropriate sampling rate (and thus frequency of clock ϕ in the delay elements of figure 4.1), the time discrete counterpart of the time continuous transfer function can be derived to

$$G(z) = A_0 A_z \frac{n_0 + n_1 z^{-1} + n_2 z^{-2} + n_3 z^{-3}}{1 + d_1 z^{-1} + d_2 z^{-2} + d_3 z^{-3}} \tag{4.8}$$

Equation 4.8 can directly be translated into SystemVerilog code by virtue of the so-called *type II filter structure* [10].

Figure 4.1 shows the generic type II time discrete filter in which a time and value continuous system with gain, such as an amplifier or a passive, lumped element filter can be converted to. In SystemVerilog, the filter weights could be implemented as fixed, numeric parameters which would then need to be known at compile time of the module. However, there are serializer subcomponents, most notably the CTLE or the VGA, which require adjustability of their transfer functions during runtime. The filter weights must therefore be dynamic values (real values/registers) rather than fixed parameters. Therefore, a numeric or analytic translation between the pole/zero representation of the time continuous view to the parameter set of the time discrete manifestation needs to be realized. While a numeric translation is indeed possible, SystemVerilog would not be the native language to implement this process - after all, its design intent is not the formulation of rather complex numeric transformation processes. As described in section 4.3, however, the openMGT extension to the opensource MATLAB clone *Octave* along with Octaves *signal package*[1] could potentially be used here to even translate time continuous functions of almost arbitrary complexity to their digital counterparts. For reasons of simplicity and due to the abovementioned limitation to only a few poles and zeros of interest, the approach

1 A package is an extension to the core functions of the numeric CAD program *Octave*. The signal package implements functions to define and analyze both functions of time discrete and time continuous realm and also allows to translate between them.

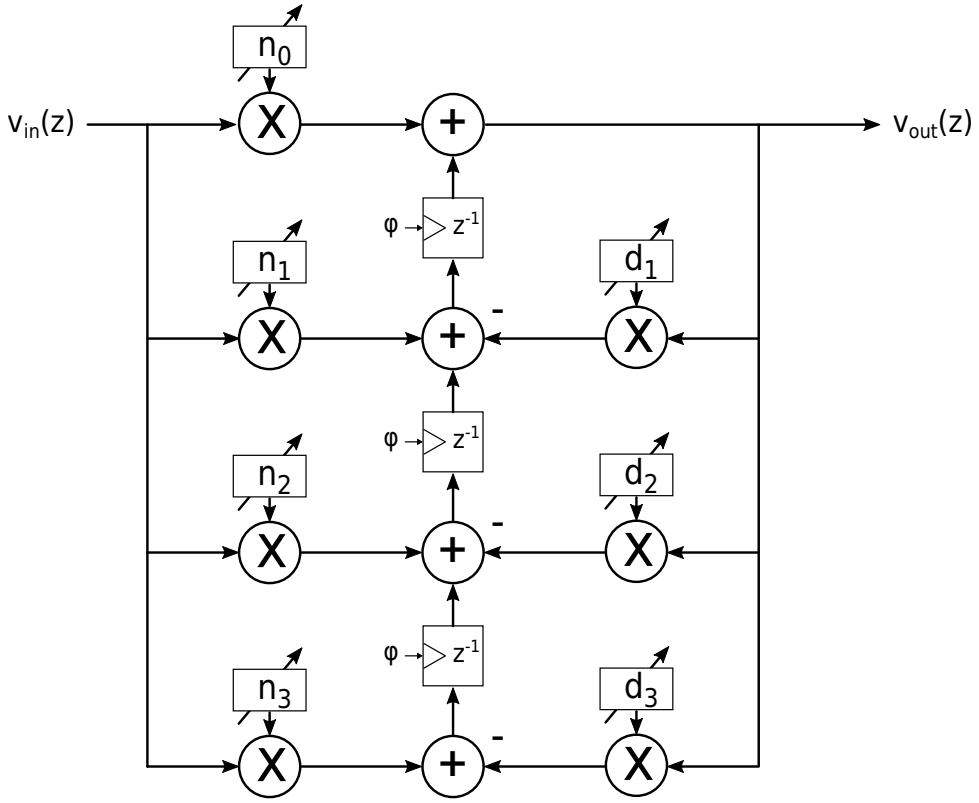

Figure 4.1: A time discrete type II filter consisting of feedforward and feedback filter weights (the gain multiplier A_z) is a part of the nominator weights n_x and A_0 not shown for clarity

taken here is to explicitly compute the translation analytically and only with respect to the transfer function of equation 4.7. This, of course, also comprises all transfer functions of more moderate complexity, i.e. with a reduced number of poles or zeros, as it is easily possible to let $\omega_{\mathrm{p/z,n}} \to \infty$. This flexibility in translation is also needed when multiple filter structures of different sampling rate T_S are needed. T_S is a fixed parameter of the filter module - the time step does not need to be changed during run time. It may, however, change with subcomponent since it is the bandwidth and equations 4.4 or 4.5 which define the required magnitude for a given accuracy requirement.

With a rather tedious and error prone process of linear transformation and substitution, we can show that the following relations between the Laplace and Z-Space parameters hold:

- for the elements in the denominator of equation 4.8 we have

$$\begin{aligned} d_1 &= \frac{3(bc-1)+b-c-a+ac}{d_0} \\ d_2 &= \frac{3(bc+1)-a-b-c-ac}{d_0} \\ d_3 &= \frac{bc-ac+a-b+c-1}{d_0} \end{aligned}$$

with

$$d_0 = (a+b)c+a+b+c+1$$

and

$$\begin{aligned} a &= T_\mathrm{S}\omega_{\mathrm{p}0}\,\omega_{\mathrm{p}1} \\ b &= T_\mathrm{S}^2\omega_{\mathrm{p}0}\,\omega_{\mathrm{p}1} \\ c &= T_\mathrm{S}\omega_{\mathrm{p}2} \end{aligned}$$

- for the elements in the numerator of equation 4.8 the translation is given by

$$\begin{aligned} n_0 &= e+d+1 \\ n_1 &= 3e+d-1 \\ n_2 &= 3e-d-1 \\ n_3 &= e-d+1 \end{aligned}$$

with

$$\begin{aligned} d &= T_\mathrm{S}\omega_{\mathrm{z}0}\,\omega_{\mathrm{z}1} \\ e &= T_\mathrm{S}^2\omega_{\mathrm{z}0}\,\omega_{\mathrm{z}1} \end{aligned}$$

- finally, the amplification factor A_z translates to

$$A_\mathrm{z} = \frac{T_\mathrm{s}}{d_0}\frac{\omega_{\mathrm{p}0}\,\omega_{\mathrm{p}1}\,\omega_{\mathrm{p}2}}{\omega_{\mathrm{z}0}\,\omega_{\mathrm{z}1}}$$

As can easily be guessed, the process of finding the set of transformation equations grows ever more tedious with an increasing complexity of the initial transfer function in terms of zero and pole count. On the other hand, for the most practical cases, the transfer function of equation 4.7 is suitable for most subcomponents in the serializer design. As the definition of this *Z-Filter* module is central to all submodules depending on it, a concise SystemVerilog testbench to assess its correctness must be defined. This self-consistent approach on formulating the model and the testbench, which will then later be used to

verify the actual implementation against the model, relies on the openMGT and the OCM framework. We will therefore pick up the thread we started here in the next two sections.

4.2 The openMGT framework

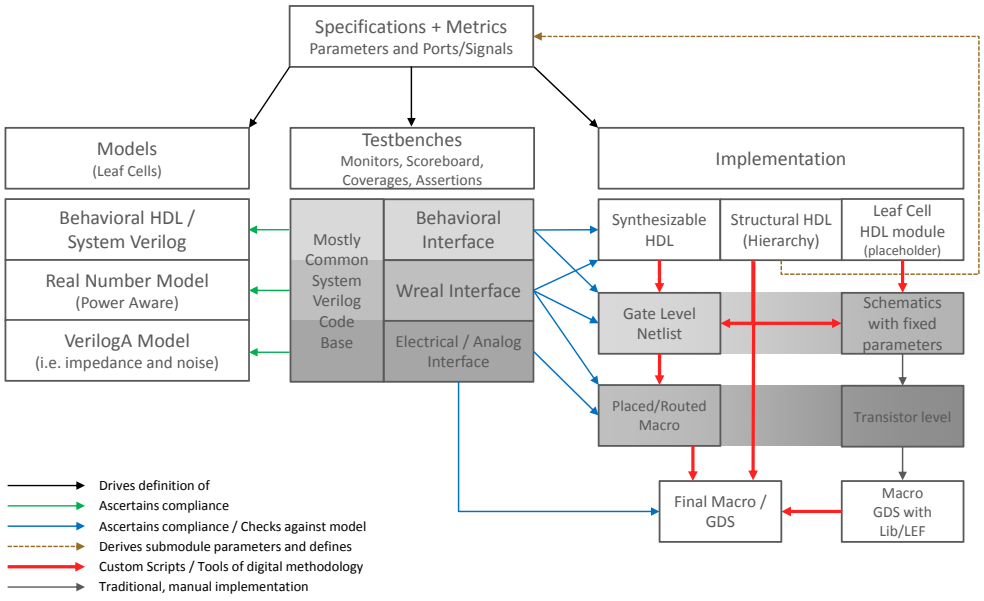

Figure 4.2: Overview of the openMGT framework

Figure 4.2 presents an overview of the openMGT framework. The framework is centered around the idea of simultaneous conception of testbenches, models and general design structure as a hierarchy of subdesigns (components or cells, we will use these terms interchangeably). The development process is a top-down specification and metric driven flow. Specifications are well-defined, observable (hence testbench checkable) quantities of design performance. For a serializer, they would for instance be given by the I/O standard specification. Metrics are essentially the same, apart from the abscence of a higher level constraint to them. Instead the constraint to a metric may originate from another performance metric or is imposed on a design by subcomponents of the same hierarchy. The common mode of a differential amplifier not interfacing with the outside world of the overall design is a good example for a such a metric. The major difference

between specification and metric therefore lies in the fact that metrics of one component may be traded off against those of another to realize a certain, higher level design metric or eventually specification. In the context of a network communication link, if we are free to define our own protocol stack and there is no standard to adhere to, the distinction between specification and metric would evidently be obsolete. In the following, we will therefore carry on with the term *metric* exclusively.

Metrics are often associated with one or two distinct ports of a component or the component itself. In addition and close relation to digital designs, components have parameters such that they may occur within a design hierarchy numerous times, possibly with a different parameter set for every so-called *instance*. In larger hierarchies, oftentimes the implementation of a subcomponent itself is just an arrangement of more specific subcomponents. In openMGT, these cells are dubbed *structural* as they can be described as a pure relation of subcomponents. This is done in ordinary Verilog HDL language. While for digital designs, these 'quantized' notions come about naturally, there is only a subset of typical analog components that is suitable for the same kind of treatment. Nevertheless, especially in mixed signal designs with a decent number of DAC and ADC based control loops, the automated procedures of defining vectors and arrays of instances simplify the description of a system compared to a schematic (i.e. graphical) view substantially. Moreover, the well established netlisting routines of HDL to gate-level compilers can be used to automatically generate large schematics, symbols and schematic templates - a mechanism that can be used to enforce consistency between modelling and implementation view [38]. A strict separation of digital and analog realms is an essential part of the workflow. This is because the digital subcomponents will and shall of course profit from the well-established industry semicustom implementation flow. Analog components on the other hand are so-called *leaf* cells, a special kind of structural cell. The leaf cells define the interface and parameters of the given subcomponent and serve as a placeholder for the particular model to be used.

4.2.1 Leaf cells

Leaf cells may then contain one of the model views as required by a particular (higher level) testbench or the actual schematic implementation itself. The translation process between the text based Verilog description and the graphics based schematic representation is performed by a custom scripting process (for details see [38]). This ensures consistency between the structural description and the schematic implementation of the leaf cell. As evident from the foregoing discussion, in order to model and test a particular metric the *model scope* has to be chosen appropriately. The various model scopes are defined in the following and presented in conjunction with a small example of how the scope may be of use in a larger design context:

Behavioral HDL Behavioral HDL relies on all constructs provided by the particular HDL language. For openMGT, the HDL used is Verilog. Since leaf cells are never going to be used in the semi-custom design flow, the full spectrum of language constructs can be employed without adhering to best coding principles for synthesizability. Care must be taken however to define HDL which simulates efficiently. Especially the number of events generated by the model and the number of constructs sensitive to events needs to be kept small for good simulation performance. Considering an amplifier within a larger design context, its main purpose from a behavioral point of view is to pass on a signal from the input to its output. Behavioral HDL, in addition to being discrete in time, only supports digital levels, the high impedance state and an error state x for a given signal/port. We can thus model some details of quite general behavior with respect to amplifier operating conditions. These include for instance the proper reaction to the presence or abscence of ground and power supplies or reference voltages and currents. A differential NMOS CML amplifier stage for instance might pull both of its inputs to supply potential when the reference voltage for its tail current source is grounded. Behavioral HDL therefore possesses its major strength in checking proper interconnectivity, its comfortable way of defining more complicated component interfaces (with tuning and calibration vectors for instance) and in asserting the correctness of the general signal flow in the design (i.e. that the output of an amplifier chain is not inverted).

Real number model As mentioned above, with real number models the concepts are limited to a signal flow description or, in case of user defined net types and resolution functions, to slightly more complicated arrangements such as some basic Kirchhoff conservation type calculations [1]. On the other hand, it is the level of abstraction that brings with it the fundamental speed-up required to solve more complicated analysis tasks of systems with large time constant spreads or to simulate designs of larger scope. Even though the signal flow view is not much different from that of the behavioral description, the availability of virtually value continuous net and register types allows to implement some very important aspects of system subcomponents. In the context of the amplifier example, it is now for instance possible to check input common mode levels, define output common mode levels, signal swings and, by virtue of the Z-Filter module described in section 4.1, even transfer function properties. With the poles and zeros even being adjustable, the central functions of linear equalization and variable gain adjustment can be described and parametrically constrained already at this fairly abstract level. It must of course be understood that there may be some reiterations needed once the actual design of a component is known. This can either mean an adjustment in metric values or the

1 An example would be the summation of currents on one real number structural component while tracking impedance or voltage on another, for well constrained circuit topologies, the problem can often be simplified in this manner

further sophistication of a model due to some unforseen "parasitic" effects. For some basic building blocks, it is also possible to define a power (or rather current) consumption estimate and use the built in cummulative summation resolution function to obtain an early estimate of the overall power consumption in the (sub-) design.

The increased simulation speed of these models then allows to explore the design space or predimension the subcomponents much more rapidly than with models of greater accuracy. Such exploration in the context of a serializer design may include tradeoffs in equalization procedures, validity of calibration routines or the impact of higher level design decisions such as line or data coding on the non-linearized (large signal) model of dynamic loops such as those formed by the CDR, PLL or the calibration and equalization loops. For these mechanisms to work well, openMGT chooses to include deterministic effects like offsets in phase or voltage resulting from mismatch as so-called *budgeting parameters*. These parameters are initially constrained by the model designer and backannotated by the schematic engineer in case they proof to be unachievable. As a consequence, the RNM hierarchy of the models can then again be used to ascertain correct system functionality even in the presence of the new, potentially elevated metric magnitude.

Electrical / VerilogAMS The openMGT modelling procedure does not fundamentally rely on the full analog capabilites of VerilogA/MS. This level of description is rather used to provide electrical (that is analog) interfaces when testbenches need to work with both the real number and schematic based view of a subcomponent. Compared to the schematic representation and depending on the level of model detail, there can be a substantial speed up with VerilogAMS models, of course. In light of the much more significant simulation speedup with real number models, however, the serializer framework focuses its modelling efforts on the time discrete signal flow view. As mentioned, the major reason for the difference in performance gain between the electrical and RNM view lies in the different methodology of the analog and digital simulator. While the RNM models only rely on the event driven digital approach, the VerilogAMS modules, depending on the language constructs used, must rely on both. The advantage of using VerilogAMS primarily lies in the possibility of having actual conservation laws and thus an impedance view associated with module ports - a circumstance which is especially useful when simulations which include S-Parameter models need to be carried out.

Schematic (BSIM), Touchstone, BBSpice Finally, the target of all the preceeding modelling effort is an all electrical model based on schematic entry. The physical design kits (PDKs) of chip foundries usually rely on BSIM-4 or PSP based transistor models which grew fairly complex over the past decades. These models are usually a precompiled part of the analog simulation kernel and may therefore not be the bottleneck to simulation speed. It is rather the procedure of iterative convergence in the analog kernel and the potentially required,

very small time steps chosen by the simulator whenever fast transients are encountered that lead to longer simulation times as a design grows in complexity. Therefore, while simulation accuracy of this layer is by far the greatest, its speed can be very low.

4.2.2 Testbenches

Wherever applicable the ports of a subcomponent or the subcomponent itself are associated with constrained metrics and the testbenches are conceived such that these constraints are captured by the testbench and satisfied by both model and implementation. The SystemVerilog extensions of recent years [1] enables this process by allowing the definition of the same testbench with multiple different levels of abstraction and the VerilogAMS language allows for the translation process between real number, behavioral and electrical modelling domains. The possibility to reuse a large portion of a common code base therefore decreases the probability of inconsistensies in how metrics are defined or probed. There are of course some limitations to this process wherever models cannot fully capture the analog reality of subcomponents, a circumstance most common as soon as impedance dependent effects come into play. Such would for instance be the case for models of operational amplifiers in feedback configuration whose loop stability explicitly depends on the output loading condition for instance. A testbench for the exemplary amplifier in this discussion can for instance be designed to probe and extract the pole and zero locations of its transfer function in the loaded condition (i.e. in the electrical case with input and output impedances considered). The procedure of how this can be done is discussed in the next section where the openMGT C/Octave modelling framework extension is outlined.

4.3 The openMGT C and Octave modelling extension (OCM)

One of the major design guidelines of the openMGT framework lies in the strict separation of deterministic and random contributions to overall system behavior. As explained, the statistical nature of random processes makes it very unlikely to capture all the important worst case conditions within a transient simulation run, especially for systems with very low bit error rate targets. Therefore, treatment of random noise contributions in voltage and phase is delegated to the OCM numeric post processing environment which also handles the linearization and calculation of the CDR closed loop transfer function and the impact of sampler nonidealities on the data captured by the receiver. Also, once the modelling process is complete or the schematic design of the serializer system is fully available, the entire design has to be simulated in concert with the physical target channels. While this task is natively handeled for schematic based designs in conjunction with S-Parameter based channels, the real number model layer performance would severely suffer from being cosimulated with a file based channel model. File based channel models are

always simulated within the analog simulator when interfaced to RNM subcomponents via VerilogAMS connect modules. However, all negative consequences of invoking the analog simulator will then again be unavoidable. Therefore, a mechanism needs to be devised by which physical channels in S-Parameter notation can be embedded within the RNM context. The OCM framework extension is also used for this purpose. As will be explained in chapter 5, even with the superior simulation performance of real number models, the time to simulate the required number of bits in order to capture the worst case response of the system channel comprised of the serializer equalization stages and the channel would be much too long for design space exploration and verification tasks. Therefore, the numeric post processing relies on a statistical algorithm to retrieve the worst case eye in a much faster way. The role of transient simulation within the openMGT framework therefore mainly reduces to functionality checks, system calibration and equalization and finally the simulation of pulse response sequences for final performance verification.

SystemVerilog code offers the possibility of extending the simulator itself with custom C code. This code is compiled at elaboration time of the simulator and hence runs within the address space of the simulator itself. The application programming interface (API) which is called DPI defines how data is handed off between the SystemVerilog and C realm [1]. How this is handled internally totally depends on the particular implementation of the simulation software, only the standard access functions in both the SystemVerilog and C space along with data type definitions are defined in the standard. There is no transparent access to the data structures of SystemVerilog from the C environment. Interactions always have to be triggered by user defined simulator functions from within SystemVerilog and only data defined as inputs or outputs of these functions can be read or written to from C code. Within SystemVerilog, functions are executed at a particular point in (simulation) time. While regular Verilog tasks may have time control statements and may thus be used to describe a sequence of events taking place over time, Verilog functions finish in the same simulator time step they are called in [19]. With respect to the modelling procedure of channels in RNM context, this is of distinct importance as will be seen in section 4.4. DPI translates the SystemVerilog data structures to a C counterpart. The ultimate goal of OCM is to also provide access to the powerful and optimized numerical algorithms of the open source MATLAB clone *Octave*. Therefore, there is also an additional C++ to Octave translation layer required, which uses the Octave API to synchronize data between these two separated realms as well.

4.3.1 General architecture

Figure 4.3 shows the implementation structure of the openMGT OCM extension. There are two separate intentions to the framework which share a common code base. The first is to provide testbenches access to numerically intense post processing operations such as

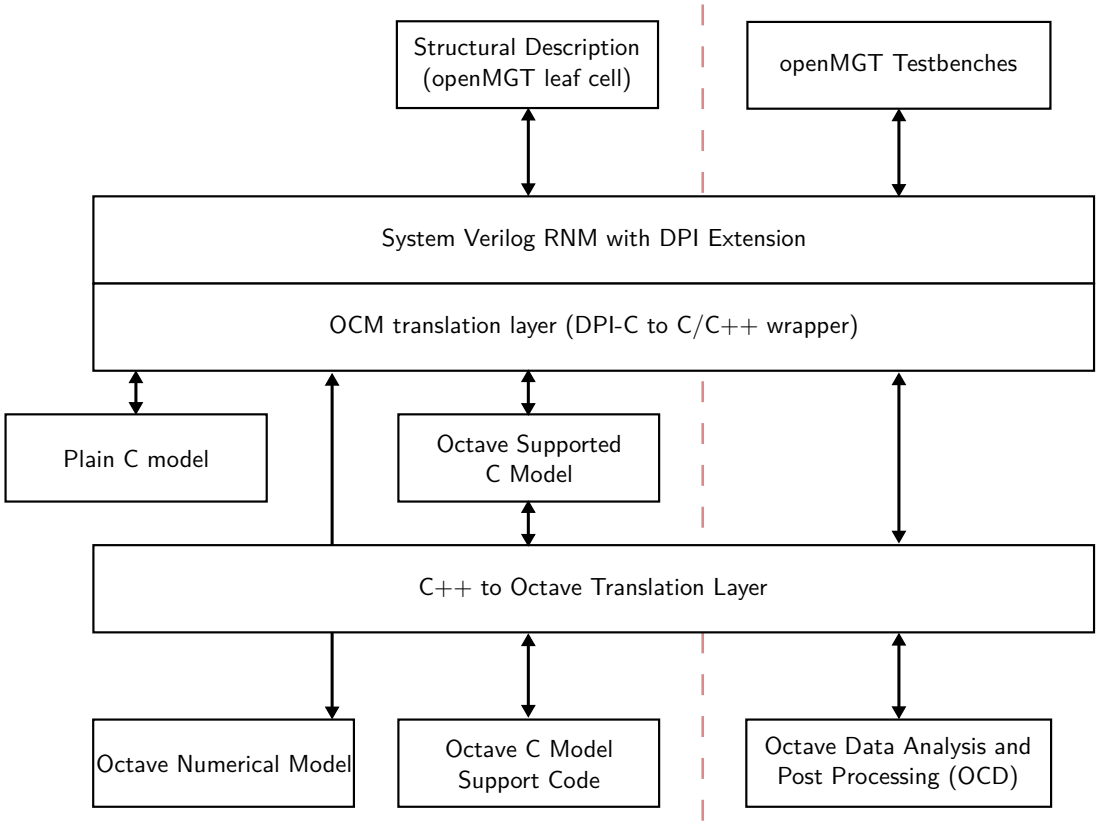

Figure 4.3: Structure of the openMGT OCM extension via DPI

required during link budgeting (see chapter 5). The other is to implement an additional real number model layer which allows to realize numerically intense operations such as convolution processes and the definition of more complex system responses which cannot be efficiently realized with the SystemVerilog data structures.

In either case, the SystemVerilog description of openMGT leaf cells and testbenches both call user defined functions as defined in the DPI extension layer. Alongside some static environment initialization and finalization functions there are also functions which allow for data hand off to and from the C and Octave domain in a flexible manner. The *ocmOctavePostProcess* command for instance only restricts the number of possible input vectors of flexible length to four while the output is fixed to a single vector of variable length. This restriction is only owed to the fact that multidimensional arrays are not yet supported by the SystemVerilog DPI standard.

```
import "DPI-C" function int ocmOctavePostProcess
(
    input string scriptName,
    input real vec0[], input real vec1[],
    input real vec2[], input real vec3[],
    input int sizeVec0, input int sizeVec1,
    input int sizeVec2, input int sizeVec3,
    output real vecRes[],
    input int sizeVecRes
);
```

The listing above shows the SystemVerilog header definition required to describe the *ocmOctavePostProcess* function in SystemVerilog context. Apart from the already mentioned vector and size arguments (note that empty square bracket indicate a dynamically sizable array which translates to a mere double pointer in C space), the scriptName argument will tell Octave which particular script to execute with the data posted by the simulator. There is no automatic mechanism of checking consistency between SystemVerilog and Octave data definitions, i.e. which data is posted in which vector. Octave script files need to be written with this circumstance and the appropriate error checks in mind while SystemVerilog code should usually not rely on the unconditional success of Octave script execution. The function return type can be used for this purpose. In this particular example, the keyword *int* allows to return a status value on which the SystemVerilog code can evaluate appropriately. Data to be used within simulation context is always passed via function arguments of type *input* and *output*. With this arrangement, testbenches can pass the data compiled during a simulation run directly to the appropriate openMGT OCM data analysis and post processing backend (OCD) post processing script and retrieve the values required for the verification task at hand.

The definition of computationally intense leaf cells follows a slightly different path. As will be described in section 4.5 for instance, Octave serves its purpose well for preparing model data or defining more complicated system responses in dependence of SystemVerilog module parameters. The actual task of computing a subcomponents response to a transient input stimulus, however, mathematically requires continuous convolution within the model. As the number of inputs and outputs to a model grows, there may even be numerous convolution processes required at once. Therefore, an efficient modelling approach to these type of problems while leveraging the performance of the computer hardware on which the simulator runs can only be achieved by implementing computationally intense module operations directly in C/C++ code. Additionally, these types of operations are usually triggered for every time step of the simulator. It is an apparent benefit to performance if the number of API translation processes per time step is kept at or below unity (see section 4.4). If Octave support is not needed at all due to a fairly simple model definition, the new modelling layer also supports implementation in plain C. Conversely, if the number of calls to a DPI model function is not set by the time step and simulation length of the simulator, models which use the C realm as a passthrough and which are entirely implemented within the Octave space can also be realized.

Another benefit of having metric driven, parameterizable subcomponents whose model is directly reflected in the Octave domain lies in the seamless access to the very same data for simulation post processing, especially in the context of overall (lane) system budgeting. Consider for instance the case of the linearized CDR model which will be needed to compute the final, recoverable eye diagram and hence the final BER estimation. The properties of the filter loop depend greatly on the properties of the receiver sampler

(see section 5.3.6). While the transient simulation asserts correct functionality with respect to the signal flow of the design using sampler metrics as defined in the sampler leaf cell of the actual design, the very same parameters will be reflected in the numeric portion of the sampler model during post processing. This is a further example of how this particular approach enforces consistency between the various simulation views of the sub- and overall design.

4.3.2 Self consistency of testbench and model

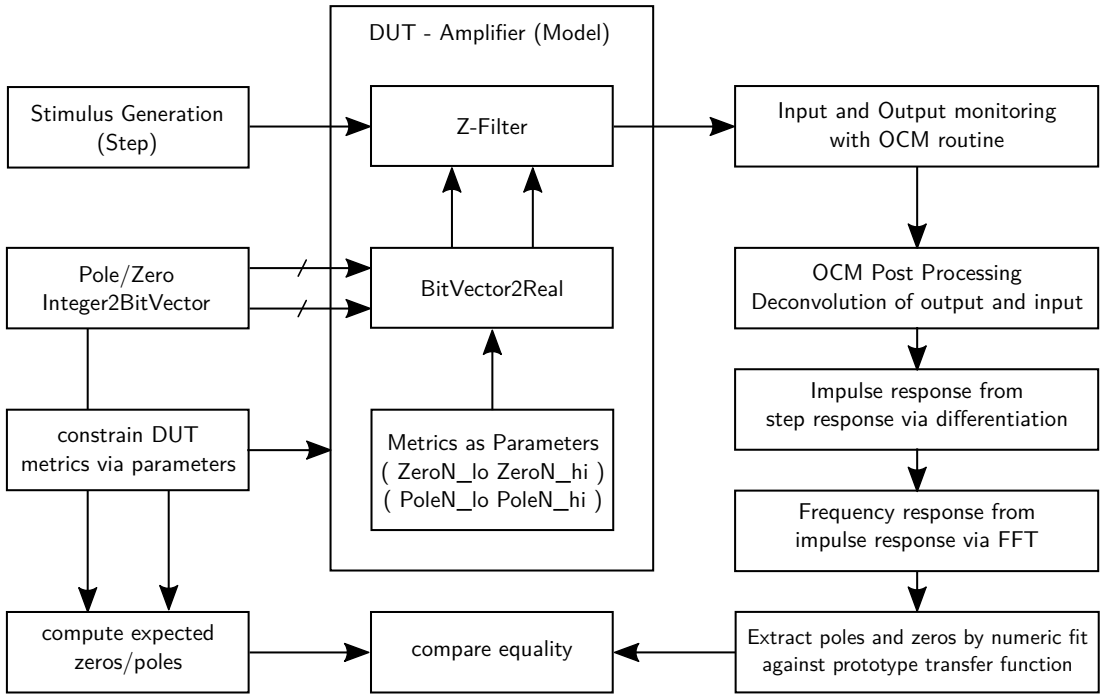

Figure 4.4: Structure of the self consistent testbench and model development for amplifier transfer function verification

As an example, an amplifier with adjustable zero and pole is modelled here. It is defined by two digital ports which translate the digital vector to real number variables which in turn are input to a Z-filter module. The amplifier has a general section where input common mode range and reference voltage range can be checked. The module receives the amplifier input data, converts it to single ended information and passes it on to the Z-Filter module. The required HDL code for the Z-Filter can readily be derived from the direct type-II filter representation. Care must be taken not to propagate don't-cares (x) into the IIR filter as this state will be irrecoverable. Therefore, as a reset and power down, resetting filter coefficients will reset the internal state. This of course is a very apparent difference between the descriptions in the analog, time continuous and the real number

domain. Whereas in the first, the filter is stateless, the signal flow description of the second must necessarily be stateful.

The general testbench can send steps, pulses or PRBS streams into a DUT. Signal monitors capture both in and output at simulator step time resolution. The signal monitors use DPI routines to transfer data to the Octave domain. Octave uses this data to derive the frequency response from the step or pulse function (differentiation of the step response in time gives the according impulse response). The numeric backend then fits the transfer function against equation 4.7 numerically. The resulting locations of poles and zeros must match the model input parameters as given by model instantiation. Since the process of creating the model and probing its properties by testbench are vastly different and the numeric domain of Octave is considered well checked with respect to implementation errors, the testbench is implicitly assumed to extract the right metric while the model is shown to define the extracted metric in the proper manner. This also includes correctness of the formulas derived for the generic three pole, two zero system of section 4.1 along with its SystemVerilog implementation. The result of the process can be seen in figure 4.5 which shows an exemplary plot of a CTLE equalization stage.

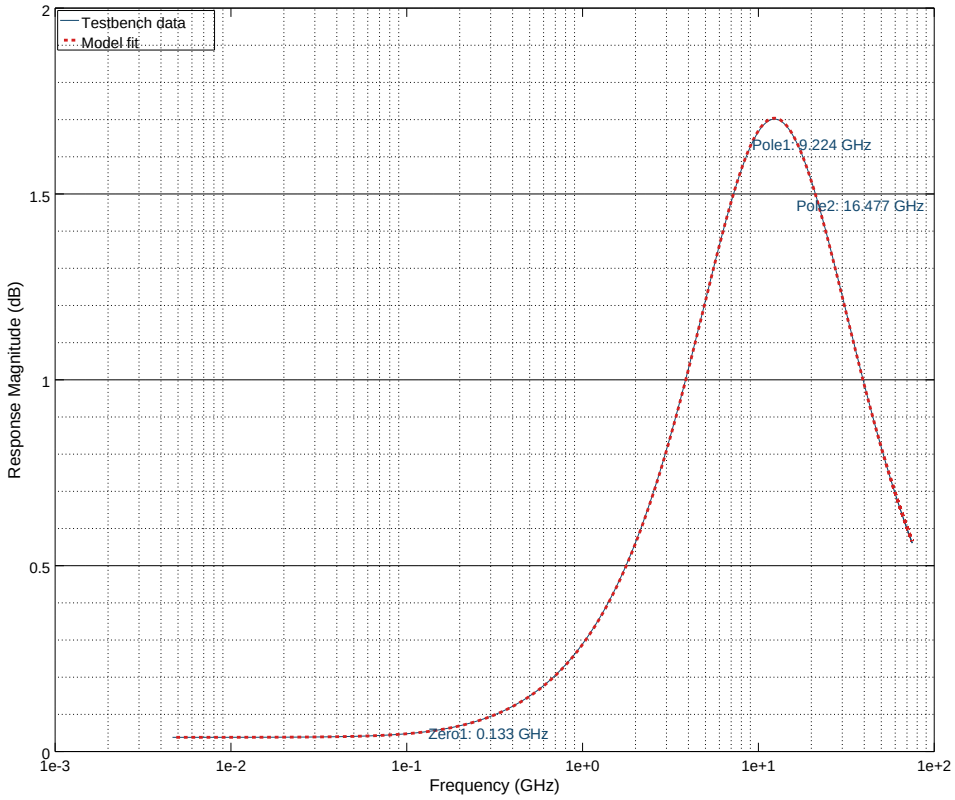

Figure 4.5: Exemplary frequency response of a CTLE structure as probed by the testbench presented here. The poles and zeros given in the figure match the instantiation parameters of the design under test.

4.4 Channel modelling process

The previous sections and chapters underlined the importance of a fast and flexible channel model for real number model based simulations, especially given the fact that the channel response is one of the most computationally intense parts of the overall simulation process. While RNM models are strictly constrained to a signal flow view, channel properties being very complex are usually described (and recorded) in the frequency domain by lab instrumentation or ECAD tools which are and must be impedance aware. Even with analytic models such as JSM, the adjustable model parameters culminate in a frequency transfer function. We thus require a transparent transformation process which keeps the channel models for serializer analysis and budgeting in RNM space consistent with the model files used for the (schematic) implementation level. In order to support fast design space explorations, the modelling structure needs to be of adjustable precision and timestep size. For instance, assessing the functional correctness of equalization calibration loops

with the two time constants of data rate and adaption loop which are spaced by a factor of rouhgly a thousand may work with fairly coarse grained simulation resolution. The actual budgeting procedure, however, will rely on much smaller timesteps to correctly estimate the magnitude of residual inter symbol interference and the effects of equalization on the CDR sampling point. Finally, due to the strict separation of random and deterministic effects in the openMGT framework, there will necessarily have to be numeric analysis steps in the OCM budgeting procedure (see section 5.3) which also need to have access to the original channel frequency transfer function(s).

The openMGT OCM channel implementation attempts to solve these requirements in the following way: The general idea is to mimic the channel as an FIR filter with its taps given by the impulse response between two ports. This approach lends itself naturally to the RNM modelling realm. It requires to prepare the data numerically given a set of *general model parameters* such as target precision or simulator time step. This once obtained data is transferred to the C extension of the simulator where a custom convolution implementation then efficiently computes the channel responses for the input stimulus provided by the SystemVerilog environment. These steps are outlined in the following in greater detail and are also laid out graphically in figure 4.6.

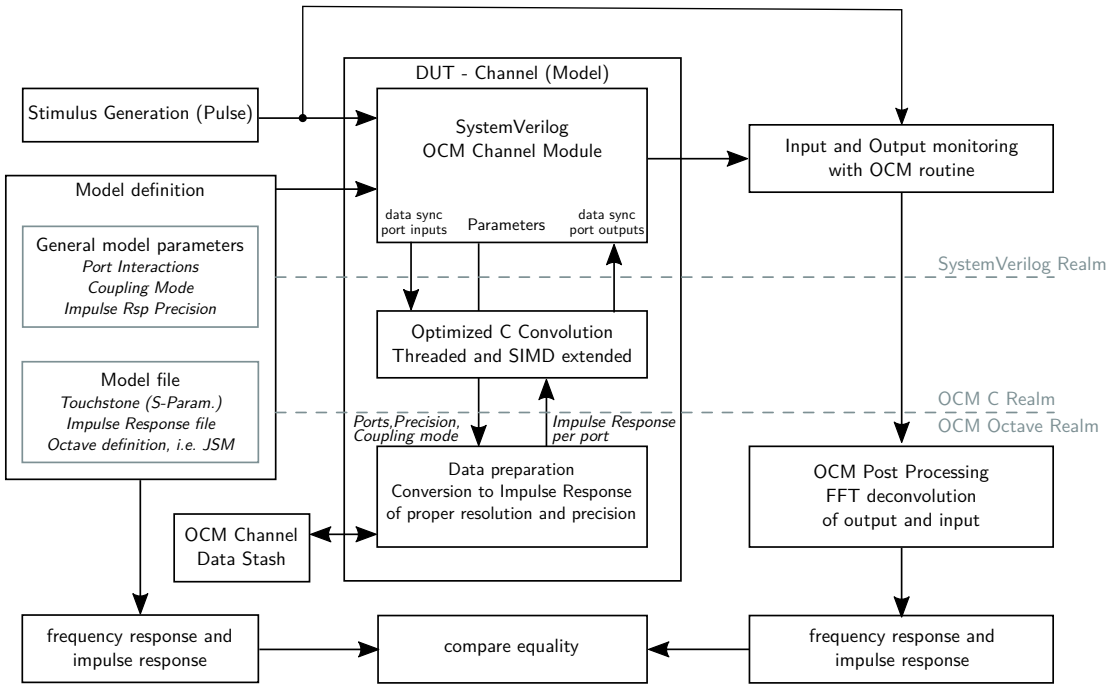

Figure 4.6: OCM channel modelling and testbench flow

4.4.1 OCM data preparation

The input data describing the channel can be of various origins which all must represent a differentially coupled transmission line with its four ports. This sets the structural requirements to the schematic and SystemVerilog channel module/instance. The differential ports are combined by defining port 1 to be the positive node of the transmitter output, port 3 to be the negative node of the transmitter output and ports 2 and 4 to be the positive and negative input of the receiver respectively.

- The source can be a Touchstone file representing the single-ended S-Parameters of the four port. This is the most native model description as it is the direct counterpart to the schematic based, direct inclusion customly supported by ECAD tools.

- The origin of the data may also be an impulse response file (i.e. from a specification). It is assumed that this response was recorded with a differential channel. Both traces between port 1/2 and 3/4 will be assigned with the identical impulse response. Conceptually, this models two uncoupled traces of half the differential impedance. Since there are no EMI sources or aggressor nets in the RNM simulation (yet), this abstraction step is considered to be permissible. Also, providing impulse response functions is a more common thing among older, less demanding and slower serial standards. Newer standards always define impedance masks in accordance to which the channel can be modelled with modern ECAD tools. They in turn then provide S-Parameter files as input to the procedure presented here.

- A third option is to describe the frequency transfer function with a numeric model. The Johnson Signal Model introduced in section 3.2 is an example of such a model. Its free model parameters can directly be specified within the SystemVerilog component thereby ensuring consistency between transient simulation and subsequent numerical budgeting.

Unless the impulse response is already given explicitly, it is numerically computed by Octave via an inverse Fourier transformation. Care must be taken with respect to the definition ranges of the input model data. The transfer functions which are usually given for the positive range of frequencies are first extended to negative frequencies. Since we assume the impulse response to be purely real, the real part of all frequency components is symmetrical about the y-axis, that is $Re\{H(\omega)\} = Re\{H(\omega)\}$ and the imaginary part must exhibit antisymmetry about the y-axis, that is $Im\{H(\omega)\} = -Im\{H(-\omega)\}$. This effectively doubles the transformation window and perturbations due to the limited window size of the discrete transformation process are reduced. The resolution and length of the resulting impulse response will of course depend on the highest frequency content of the transfer function and on its resolution (refer to equations 2.2 and 2.3 in section 2.1).

Usually, start frequency f_s and resolution f_r of ECAD or lab instrument provided data is the same and sets the length of the computed impulse response. Since we are most interested in the high frequency properties of the transmission channels, the constraints on f_s itself are fairly moderate as long as the propagation delay of the channel and therefore the round trip time for resonance induced ISI is not too large. A range between $5 - 50\,\mathrm{MHz}$ is usually sufficient. As far as f_r is concerned, however, a low value around $5 - 10\,\mathrm{MHz}$ is preferable in order to capture the various potential resonances in the frequency transfer function. The stop frequency f_e on the other hand needs to be chosen such that f_e is well beyond the Nyquist frequency $f_{Ny} = \frac{1}{2T}$ of interest. This allows a good impulse response resolution to capture the various discontinuities in the channel for a correct prediction of ISI.

In order to find a reasonable constraint for f_e, section 3.4 about the PSD of NRZ test patterns provides valuable insight. It can be seen that the main lobe of power in the signal drops to zero at $\omega = 2f_{Ny}$. There is a second lope starting at this point and reaching to $\omega = 4f_{Ny}$. However, its power is just a fraction compared to the main lobe (five versus ninety percent, which can also be seen from figure 3.23 in section 3.4). Furthermore, channels with strong attenuation already at the Nyquist frequency will diminish this fraction of spectral power content even further. The highest frequency content in the data provided to the OCM framework may therefore be limited to twice the Nyquist frequency of the highest data rate to be supported by the serializer system.

The channels impulse response taken directly from the inverse Fourier transformation might be a very long vector. In order to comply with the simulator time steps which can be tuned to trade off precision for performance, the numeric backend may additionally have to resample and truncate the impulse response. The resampling process needs to assure that the power integral of the impulse response

$$P(x) = \sum_{n=0}^{N-1} \|x[n]\| \cdot \Delta t$$

stays constant. Naturally, the contribution per tap will therefore scale down with the number of taps per unit time. If the simulator time steps are small compared to the response duration the resulting vector of double precision values will be very long. From a convolution standpoint, however, the important section even of a very dispersive channel is fairly short. In order to decrease the vector dot product operations (to which the convolution essentially boils down to) the FIR tap vector should be as small as possible. This is what is meant by *impulse response precision*: the general model parameter describes how large the magnitude of the impulse response at a certain sampling point (tap location) must be relative to its maximum so that it is considered relevant. Defining the parameter of impulse response precision relative to the response maximum allows to support different simulator

time steps without changing model parameters.

The truncation procedure clips the impulse response from start and end towards the pulse and stops once it encounters a relevant tap. For long channels with strong reflections, this procedure may still lead to fairly long responses from pulse peak to end. On the other hand, the large propagation delay of long channels will mirror in a long sequence of quiesce leading up to the impulse peak. In contrast to the tailing sequence, the lead in sequence can, however, not simply be truncated since these taps reflect the propagation delay of the physical channel.

On high performance computer hardware, the transformation and resampling processes do not take up much time. Octave uses advanced open source libraries to implement these procedures in a very efficient way. Nevertheless, during development of system components which rely on a channel model to be tested with, regenerating the model data on every simulation run would proof tedious after a while. Therefore, if the general parameters of a channel with a given Touchstone file are not changed in the SystemVerilog code (which is the location they are defined at, see below), the required impulse responses are drawn from a disk storage area called the *OCM stash* to avoid unnecessary recomputation.

4.4.2 Optimized C convolution routine

The C channel convolution routine receives its parameters from both the SystemVerilog and the Octave OCM domain. While SystemVerilog derlivers all general and model parameters for the channel instance, the C routine passes all parameters required for impulse response generation on to the OCM domain. In turn, it receives the truncated impulse response. The noncontributing delay taps of the channel propagation delay are truncated from the main impulse response, too, but are also reported to the C routine. It will allocate a corresponding amount of memory for buffering up and delaying the stream of real number values incident on the SystemVerilog components ports. With a sequence of further general model parameters, the C convolution code and the data preparation of the numeric backend allow the explicit definition of which *port interactions* described by the model file will actually be considered for simulation. A real number simulation with a long, dispersive channel with weak coupling between the traces may for instance be plainly modelled by its forward S-Parameters S_{21} and S_{43}. If coupling in forward direction (from TX to RX) between the traces is to be considered, S_{41} and S_{32} may for instance also be of interest. The general parameter named *coupling mode* on the other hand allows to numerically compute the mixed-mode S-Parameters of the coupled structure which could then again be used to reduce the number of port interactions to merely two again (S_{dd21} twice for both traces of the differential pair channel model).

In order to speed up the convolution procedure, three distinct optimization concepts have been implemented in the custom C code:

- A ring buffer per port interaction to model initial channel delay and avoid memory copies. Apart from being far more efficient than a SystemVerilog real value delay chain, it also allows to accept a flexible number of data samples per API call (see the subsection below).
- The vector dot product operation for convolution is implemented using specific SIMD instructions.[1] It is also implemented in a general way with floating point values of double precision to support all CPU types. The degree of architecture independent GNU C Compiler optimization flags has been tuned and analyzed for best performance.
- There is one POSIX thread spawned per requested port interaction. This essentially parallelizes an otherwise single threaded SystemVerilog simulator and therefore uses the performance potential of modern hyperthreaded or multicore machines to a much better extent.

The role of the OCM C code is actually twofold. During initialization it accepts data from the SystemVerilog domain and triggers the Octave backend data preparation process from which it in turn reads back the impulse responses and the number of delay taps. During simulation, it then distributes the incoming stream of data to the appropriate threads, gathers their results and hands the data back accordingly.

4.4.3 SystemVerilog considerations

Conceptually, the SystemVerilog channel module ports are input and output at the same time. The port interaction definitions which are a part of the general model parameters contained in the SystemVeriog code, permit to define a port as both source or destination and even both at the same time. This is also necessary because of the S-Parameter return losses S_{NN} which can then be included in the modelling process as well if need be. During a simulator time step, the current voltage levels, represented by a real number on the modules four inputs, are saved by the module into a common array of four real items. The module then executes the optimized OCM channel convolution function with a SystemVerilog DPI call thereby passing over control to the custom C code extension. The new data items are passed by using the common real array as the functions first argument. In the C domain, it can be accessed by a simple double pointer and can thus be written into the respective ring buffers as described above. The collected results from the various threads are combined for each destination port and written back to the SystemVerilog simulator. This is done by specifying a second real array common to all ports and passing it as the second argument of the convolution function call. Once the DPI function call returns, the values in the

1 On the Intel machines used in this thesis, these are AVX or SSE instructions

second array can be passed on to the respective module ports by continuous assignment. The following code listings show the SystemVerilog definition of the channel convolution function as well as a fraction of the SystemVerilog model code to more accurately convey the whole picture.

The SystemVerilog DPI call is defined in the following way.

```
import "DPI-C" function void ocmChannelConvolutionStep (
    input real portsIn[`OMGT_OCM_SYNC_WORDS*4-1:0],
    inout real portsOut[`OMGT_OCM_SYNC_WORDS*4-1:0]
);
```

The SystemVerilog module parameters and real input arrays allow direct manipulation of the specific model to be used, its options (such as source file), further arguments to a numeric model (such as those required by the Johnson signal model) and of course the general parameters of the modelling process which define model accuracy and the required port interactions (for an explanation of the sync word mechanism, see below).

```
module MGT_MODEL_CHANNEL #(
  parameter string  MODEL_TYPE     = "", // SXP or JSM
  parameter string  MODEL_DESC     = "",
  parameter int     NUM_MODEL_ARGS = 16, // number channel model arguments
  parameter int     NUM_PARAMS     = 16  // number of general model params
)
(
  input   wrealsum  IN_N,   // aka. port 3
  input   wrealsum  IN_P,   // aka. port 1
  output  wrealsum  OUT_N,  // aka. port 4
  output  wrealsum  OUT_P,  // aka. port 2
  inout   wreal     REF,
  input   real      params[NUM_PARAMS-1:0],
  input   real      mdlArgs[NUM_MODEL_ARGS-1:0]
);
...
initial begin
  cid = ocmCreateChannel(MODEL_TYPE, MODEL_DESC, params, NUM_PARAMS, mdlArgs, NUM_MODEL_ARGS);
   ...
   forever begin
   ...
   if(i == `OCM_SYNC_WORDS ) begin
      ...
     ocmChannelConvolutionStep(port_hist_in, port_hist_out);
     ...
   end
   ...
 end
 ...
endmodule
```

This arrangement evidently creates a situation in which it is very likely to have at least two drivers on a single wire. The first may for instance be the driver modelling transmitter output voltage contribution, the second would be the reflected voltage of the channel port. Therefore, the module ports are defined as being of type *wrealsum*. Wrealsum is a builtin SystemVerilog wire type whose resolution function simply adds the contributions of all drivers attached to a net which matches very nicely with the principle of superposition of electromagnetic waves.

At this point, one might question the sensibility of this approach by arguing that decent ring buffer and filter structures can of course be implemented by Verilog and thus Sys-

temVerilog as well. Also, as explained in section 4.1 it is of course possible to create FIR structures entirely in System Verilog alone. However, for long, dispersive channels with strong reflections, the impulse responses tend to become fairly long. Additionally, the FIR tap weights would need to be initialized in accordance with the calculated impulse response - a step which would necessitate the extension of the resulting model with some sort of DPI call back as well. Furthermore, a flexible number of port interactions would need to be realized by cumbersome *generate blocks* or precompiler flags. The only potential way to circumvent these pitfalls would be to use a scripting language which generates specific SystemVerilog modules with hardcoded FIR tap weights from a given model file and under consideration of general model parameters defined in an additional file. Also, the resulting module would solely be restricted to the specific simulator time step resolution at hand and would require regeneration whenever the resolution was changed. Last but not least, there is no real number vector multiplication support in SystemVerilog2012 whatsoever. The FIR structure would need to be written down (or generated) explicitly. Depending on the number of required port interactions, there would be a considerable amount of real registers, adders and multipliers involved to mimic the process of convolution. Given that current SystemVerilog simulators are still single threaded programs, parallelization or any machine dependent optimization would of course not be possible at all.

Therefore using the DPI extension with its vector access functions results in the fastest and most flexible method of modelling transmission channels in a signal flow based view for RNM simulations. There is, however, a slight drawback to this solution if implemented exactly the way described above:

Every DPI call triggers a sequence of simulator intrinsic mechanisms which take care of translation processes and data consistency checks. If the C code executes very fast compared to the DPI call itself, this produces an intolerable amount of overhead. Especially if optimization schemes such as SIMD instructions or threaded execution are to be of any benefit, the translation process between SystemVerilog and C domain must be diminished in its temporal extent compared to the convolution computation time. This can actually be achieved by data aggregation. Instead of calling the convolution function on every time step, data is collected for a parameterizable amount of time and transferred to the C domain at once. The same function call also returns the computed values - the synchronization words - for the future time steps.

For port interactions whose impulse response displays a considerable amount of delay (that is, a sequence of taps below the significance threshold), the amount of synchronization words can simply be deducted from the required amount of delay taps and the synchronization mechanism will have no apparent effect on the correctness of the channel port interaction. For return losses S_{NN} however, this procedure is of course not accurate since input stimulus and resulting reflection are physically immediate. In the context of serializer

design space exploration, however, we are mostly interested in the transfer properties of the channel. Real number models do not capture the notion of impedance and it would certainly be a challenging task to model the multiple reflections occuring at a severely mismatched port with strong return loss within RNM context. For final performance evaluation of a system implementation, one may also revert back to true S-Parameter based simulation in order to capture these impedance related effects. As far as the analysis and verification targets of real number modelling are concerned, however, the ansatz presented here provides enough accuracy and captures the most dominant effects.

As can also be seen from the module code listing above, the channel initialization must also be initiated by the specific channel instance. The module parameters, both model specific and general, are passed to the numeric backend which computes the required impulse responses accordingly and triggers the memory allocation processes in the C domain. This mechanism allows to realize all required flexibility in the numeric backend including adjustable simulator time steps and therefore accuracy. It also ensures that the channel model used for post simulation budgeting will work with the same model data as its preceding transient simulation (see section 5.3).

4.4.4 Testbench and performance comparison

Following the general design idea of the openMGT framework, the channel testbench needs to assert correctness of the channel model presented here. It also needs to proof reasonable equivalence of the real number model signal flow view and the schematic representation, that is, the vendor implementation of S-Parameter integration in analog transient simulations. The arguably complex modelling scheme may lead to implementation errors at various different levels. Surely, the numeric backend data preparation is based on mature code as Octave itself is based on well-established open source libraries (i.e. the openBLAS library for linear algebra or the fftw library for Fourier transformation). Nevertheless, correct data preparation such as resampling are prone to result in scaling error if handled incorrectly. More importantly, it is the custom C convolution routine based on SIMD instructions which needs to be tested thoroughly.

To this end, a pulse response testbench is used. The stimulus generator is written in SystemVerilog. It directly stimulates the transmission ports 1 and 3 of the test channel in the real number model case. For schematic based simulations, a connect module is responsible for correct conversion to behave like a voltage mode driver of well-defined output impedance. The termination of the output ports 2 and 4 simply mimics the DC voltage division in the RNM case whereas in the schematic based simulation, the termination module is swapped for a schematic based implementation. Both input and output of the channel module are recorded over time by SystemVerilog modules called *monitors*. They, too, use an OCM routine to transfer the recorded data to the numeric backend once the

simulation finishes. In the numeric backend, the post processing routine is triggered which uses Fourier transformation based deconvolution to *numerically reconstruct* the impulse response of the DUT. The recovered and original model impulse responses can then be compared for equality. For improved visual comparability here, the impulse responses are again Fourier transformed to present a more intuitive picture of the frequency transfer functions magnitude plot.

This closes the circle of model and testbench development. Exemplary results for this procedure are given in figures 4.7 and 4.8 for the analog schematic based and real number model based representation of a CEI28-MR channel respectively.

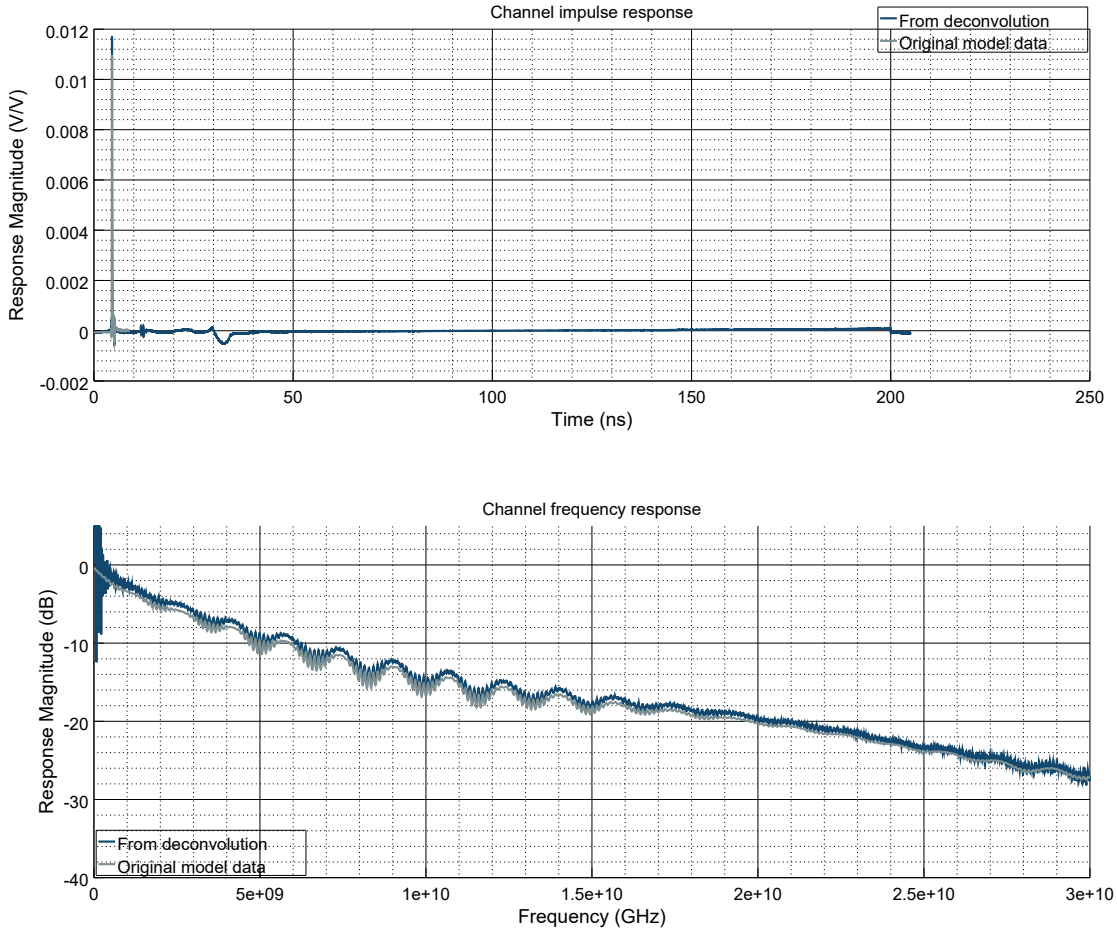

Figure 4.7: Impulse response and frequency transfer function of the deconvoluted full analog simulation together with the original S-Parameter data

From figure 4.8 it can be seen that the RNM modelling and simulation process produces a good agreement (apart from a mild, pessimistic DC offset) with original model data when a simulator time step and therefore FIR step resolution of 1 ps is chosen. The vendor supplied analog convolution process on the other hand indicates an optimistic DC offset

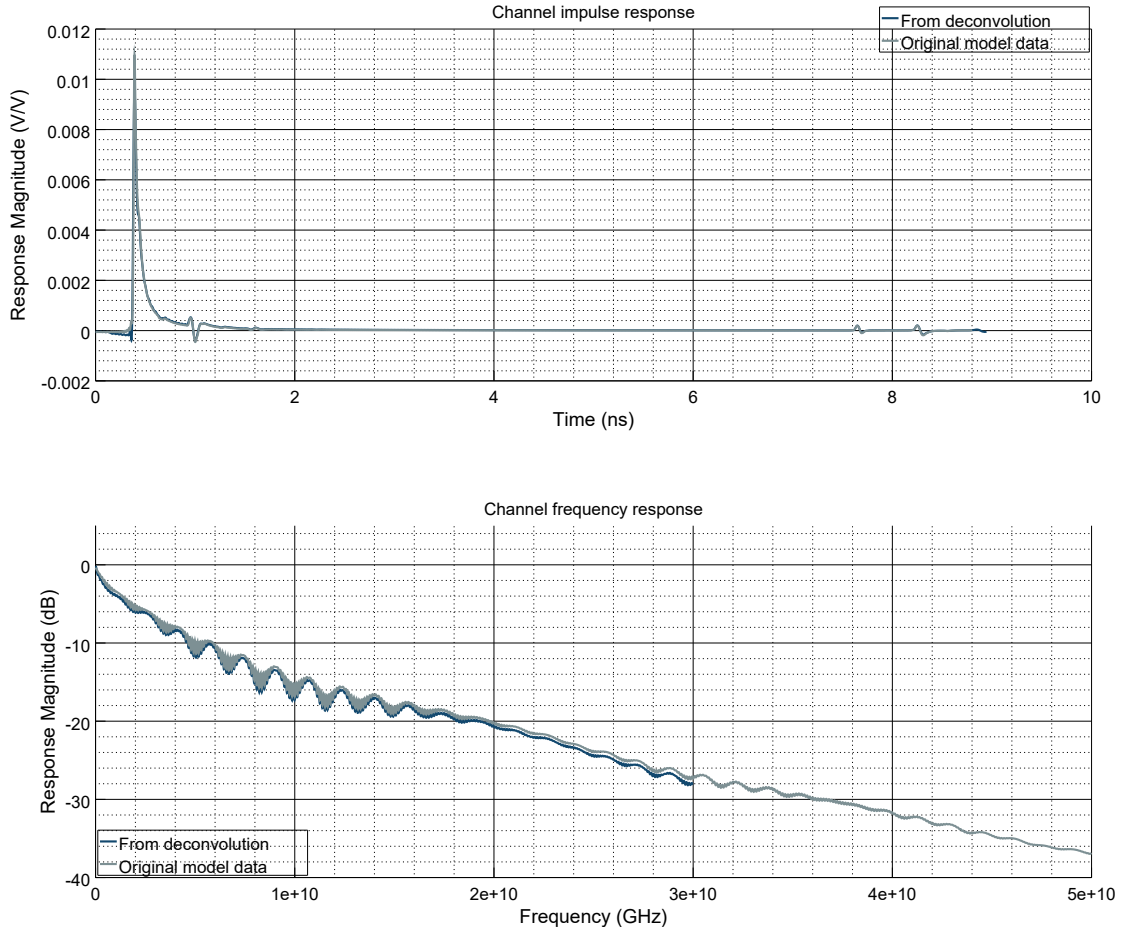

Figure 4.8: Impulse response and frequency transfer function of the real number channel model with two port interactions in coupled mode and time resolution of 1 ps together with the original S-Parameter data

and exhibits severe non-passivity issues towards lower frequencies even though a passivity enforcement parameter is supplied by the schematic instance and activated in the design. Nevertheless and as expected, the general match to the model data is very good for the bulk of the frequency range observed here.

For performance comparison the pulse stimulus of the testbench is replaced by a PRBS-31 pattern at 20 Gbps with rise and fall times of 10 ps. The simulation is executed on RAM disk and *no* simulator trace results are written to the usual waveform database so that potential I/O dependent effects are minimized. The total simulation length was set to 1 us. The simulations were carried out on a 12 core Intel Xeon E5-2630 running at 2.3 GHz with 96 GByte of RAM. The commercial ECAD vendor tool used for analog simulation features parallelization support for its analog simulation kernel. However, the software decides autonomously if it deems it neccessary to use more than a single thread. Due to the simplicity of the all-analog section of the testbench presented here, the software package chose to deactivate multithreading support. The analog simulation was carried out with

moderate error presets (which governs the time, voltage and current resolution as well as the convergence thresholds of the analog solver) and there was no maximum time step constraint given, i.e. the analog solver was allowed to freely decide how large its next time step was going to be. In contrast to RNM simulations, this particular feature can give a performance advantage for certain designs with a small to medium time constant spread. A first overview of the results are presented in the table below:

Domain	Time step	Port interactions	Sync Words	Simulation Run Time
Analog	Variable	all (16, non adjustable)	N/A	45 m47 s = 2747 s
Real Number	1 ps	4(S_{21}, S_{43}, S_{23}, S_{41})	1	26.2 s
Real Number	1 ps	4(S_{21}, S_{43}, S_{23}, S_{41})	16	17.6 s
Real Number	1 ps	**2***	1	15.2 s
Real Number	1 ps	**2***	16	9.3 s

Table 4.1: Comparison of all analog versus real number channel model simulation performance.

The baseline for performance comparison is given by the analog simulator which requires three quarters of an hour to complete the simulation task. It must of course solve its set of equations for all 16 port interactions contained in the S-Parameter matrix. Its freedom in choosing time steps as required is not very beneficial due to the high volatility of the PRBS-31 data pattern. It is unclear if the convolution process for modelling the frequency parameters is in any way parallelized. System administration tools indicate two running threads, potentially one for the analog and one for the digital simulator kernel.

The real number model simulation of comparable accuracy (albeit without the return loss parameters which are of no real benefit to signal flow simulations anyway) with an FIR filter resolution in accordance with the simulator step size and its four threads requires just about half a minute for the same task. Using the previously mentioned sync word mechanism to minimize the number of DPI function calls by transferring 16 data items every 16 time steps already pushes the run time below the twenty second threshold. However, the benefit is twofold. The reduced number of API calls reduces the overhead compared to computation. On the other hand, the greater amount of data increases the time spent on parallelized computation compared to the sequential synchronization sequences in the code.

This, however, is not the end of the line. Using the coupled mode mechanism which combines the four port interactions into its mixed mode S-Parameter representation reduces

the number of threads by two. It is denoted by **2*** in the table and uses S_{dd21} for both traces of the differential channel. S_{dd21} considers four distinct port interactions according to

$$S_{\mathrm{dd21}} = S_{21} - S_{23} - S_{41} + S_{43}$$

Even without the sync word mechanism, this improves the total simulation time to a quarter of a minute. The sync mechanism can reduce simulation runtime even below ten seconds - all of this without apparently sacrificing simulation accuracy for the physical quantity of interest: the output voltage across the termination resistors at the receiving side of the channel.

This first overview proofs the potential of the modelling ansatz of OCM. It is now interesting to find the appropriate parameters for the best accuracy performance tradeoff. Since the number of threads cannot be reduced further, the time step parameter as well as the number of sync words remain for optmization. Table 4.3 and figure 4.9 show simulation duration in dependence of FIR time step resolution and present a drastic drop in simulation time already for a mild increase in time step magnitude.

Time step	Simulation Run Time
1 ps	15.2 s
2 ps	4.5 s
5 ps	1.9 s
10 ps	1.2 s
20 ps	1.1 s

Table 4.3: Comparison of real number model channel model simulation performance versus FIR model time resolution for a single sync word and **2*** port interactions.

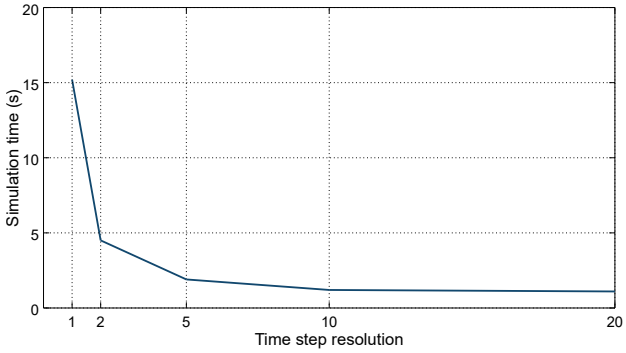

Figure 4.9: Simulation Performance versus time step magnitude: Graphical representation of the table values to the left.

The model resolution parameter affects the number of computational steps, the depth of the FIR filter as well as the number of DPI function calls all at the same time. As mentioned in section 4.1, from equation 4.5 it can be seen that time steps of 5 ps are already enough to model highly dispersive channels accurately up to frequencies of about 20 GHz and that for relaxed accuracy requirements, even less is acceptable. The most interesting question therefore is, how precise the frequency response still is if an aggressive setting of 10 ps is chosen.

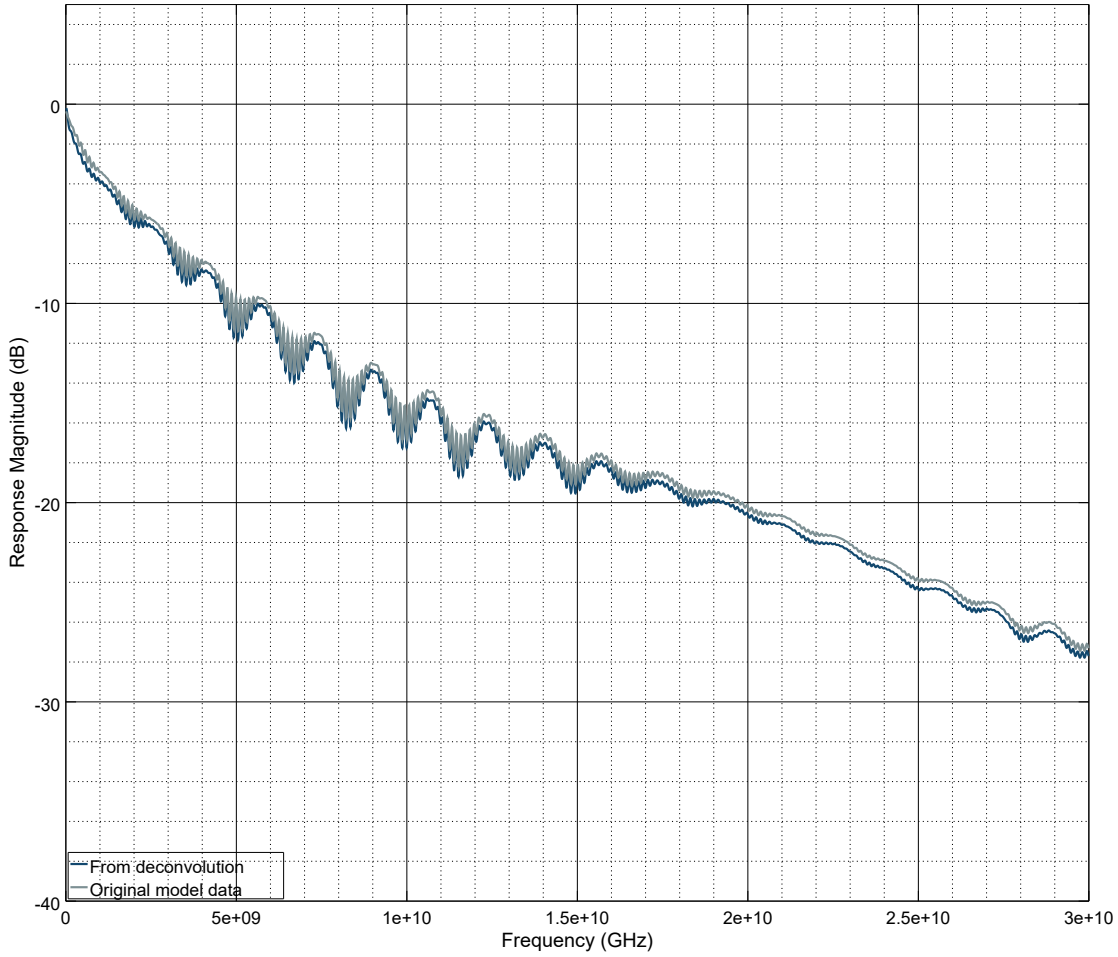

Figure 4.10: Impulse response and frequency transfer function of the real number channel model with two port interactions in coupled mode and time resolution of 10 ps together with the original S-Parameter data

As can be seen from the resulting simulations, a time resolution of 10 ps still captures the frequency response (apart from the above mentioned DC offset) very well whereas 20 ps are much too coarse to resolve the impulse response appropriately (compare figures 4.10 and 4.11). Given that the performance gain for much coarser time resolution than 10 ps is very marginal anyway, simulations with long runtimes which can tolerate poor waveform resolution can resort to this specific FIR time step.

Now that the time step parameter is analyzed, the optimal number of sync words still needs to be found. Table 4.5 and figure 4.12 therefore display simulation duration in dependence on the number of sync words used for a time step resolution of 1 ps and **2*** port interactions.

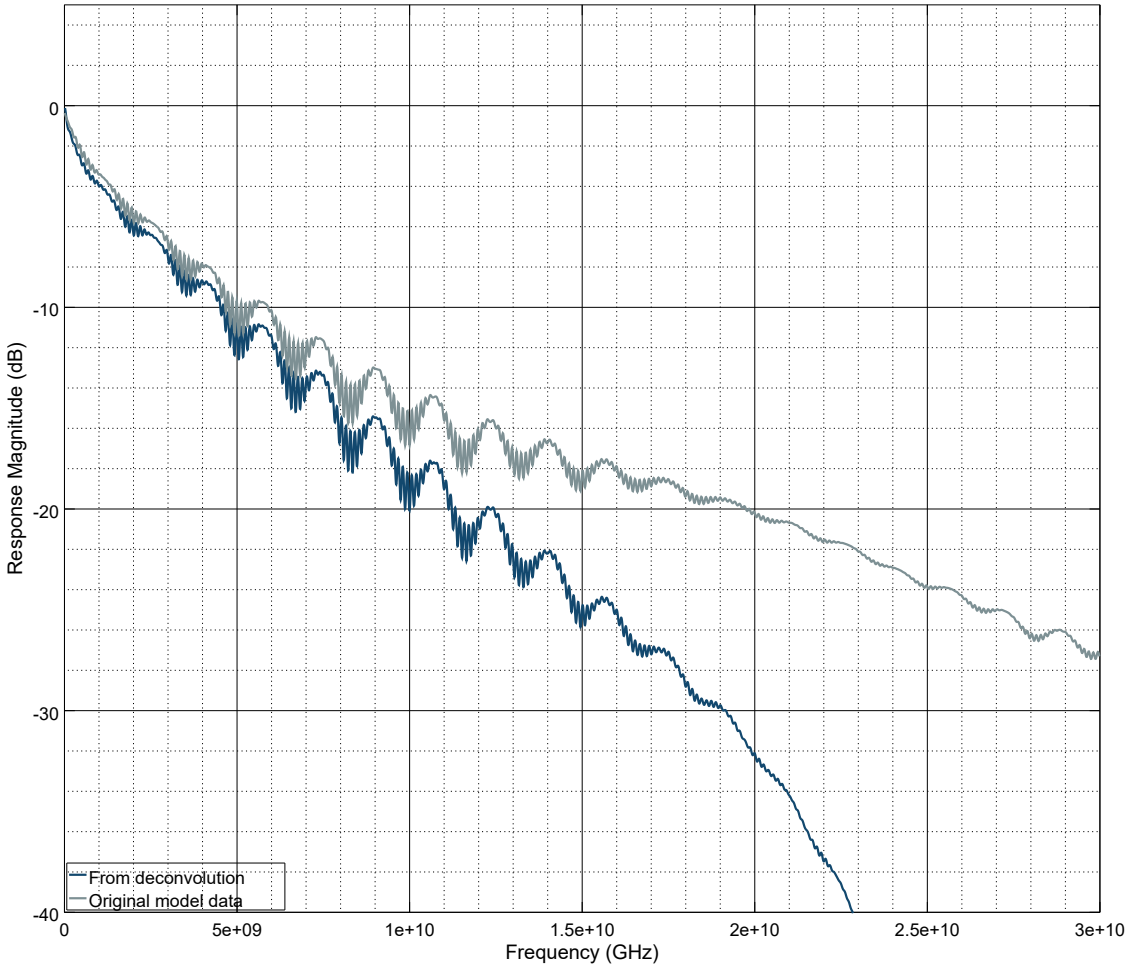

Figure 4.11: Impulse response and frequency transfer function of the real number channel model with two port interactions in coupled mode and time resolution of $20\,\mathrm{ps}$ together with the original S-Parameter data

Sync Words	Simulation Run Time
1	15.2 s
2	14 s
4	14 s
8	9.8 s
16	9.3 s
32	9.3 s
64	9.3 s
128	10.1 s

Table 4.5: Comparison of real number channel model simulation performance versus number of sync words for a time step of $1\,\mathrm{ps}$ and **2*** port interactions.

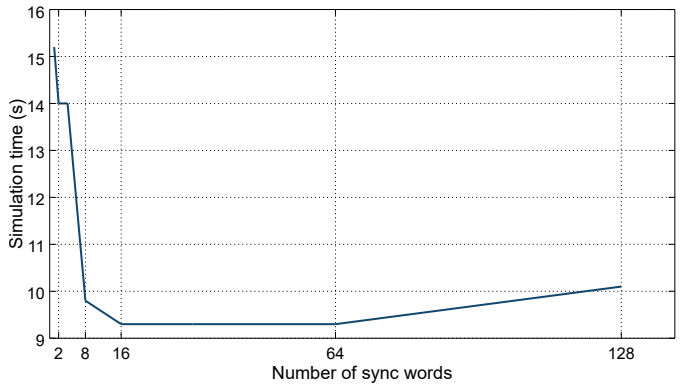

Figure 4.12: Simulation Performance versus number of sync words: Graphic representation of the table values to the left.

For the number of synchronization words between C and SystemVerilog domain, there seems to be a sweet spot for word counts between 16 and 64. Since there are four ports, the number of transmitted real values is actually between 64 and 256 per direction. Since the SystemVerilog real values find their C counterpart in floating point values of double precision which are represented by eight bytes on the x86 64-bit architecture, the number of accessed bytes in either direction are between 1 to 4 kByte. The latter is the page size of the Linux operating system which could be one reason why performance starts to drop again with an increase in sync words. It could, however also be simply related to the way in which simulator internal data structures are defined.

The best overall choices considering both optimization processes are thus given in table 4.6 and will be used for the equalization and calibration procedures within the openMGT design and budgeting framework. A resolution of 1 ps is used whenever budgeting efforts play an important role within the simulation context and slower simulation speed is accepted in favor of a better signal transition resolution. A resolution of 5 ps with 16 sync words will be used for simulations with large time constant spread such as equalization training loop analysis. For these simulations, the presented modelling procedure offers a performance factor of **1830** compared to traditional all analog simulation with moderate accuracy presets.

Time step	Port interactions	Sync Words	Simulation Run Time
1 ps	**2***	16	9.3 s
2 ps	**2***	16	3.7 s
5 ps	**2***	16	**1.5** s
10 ps	**2***	1	1.4 s

Table 4.6: Optimized real number channel model simulation performance

4.5 Sampler modelling process

In the receiver of a serializer system, the samplers mark the location where the transition from analog to digital domain occurs. Sampler performance is most vital to metrics of various serializer subcomponents and the overall performance itself, examples being the phase detector gain in the clock data recovery loop or the first tap timing loop of the decision feedback equalization. A sampler must resolve an analog voltage level of varying magnitude at its input to a digital, rail to rail signal at its output. This resolution occurs in relation to a provided timing reference, the sampling clock. In this text, the sampler is defined to sample at the rising edge of the clock and produce the digital output data explicitly retimed to the rising clock edge, too. A sampler usually consists of more than one subcomponent depending on implementation type (see below). The first, all-analog stage of the sampler is called *comparator*. In the single ended case, it compares a signal against a defined voltage threshold and produces a binary decision output. For differential signalling, the two traces of the differential pair are compared against one another to produce the decision. The decision can then be resampled by a succeeding latch to provide proper retiming with low clock to output delay for subsequent stages.

In the past, there have been comparator implementations which relied on operational amplifiers with very high gain. As these stages consume a lot of power, may produce a considerable amount of noise and are very limited in bandwidth (as the gain-bandwidth product of any technology is fundamentally limited), comparators based on nonlinear amplification are virtually the only implementation concept present in modern serializer design. The design space as spanned in the literature really boils down to two distinct implementations with minor variations: the CMOS based StrongArm comparator (as originally conceived for the StrongArm Latch [37]) or a CML based version.

CML Figure 4.13 shows the schematic structure of a CML comparator. It consists of a clocked amplification stage and a cross couple pair of NMOS transistors which realize the nonlinear gain. On the rising edge of the clock, the linear amplification stage gets turned off in favor of the regeneration section. The period during which amplification occurs and the cross coupled pair is turned off is also called the *track phase* whereas the period during which the clock signal is high is called the *hold phase*.

The major advantages of this circuit lie in its simplicity and good power supply noise rejection as a result of the CML provenance. It also provides small signal, linear amplification in addition to the nonlinear regeneration. Its major drawback is the incomplete level recovery which reaches from supply voltage only down to the CML logic low level. This low level is defined by the tail current as set by M_5 and the pullup resistor size. A second, CMOS type stage is therefore required to restore the full digital swing which adds to the power budget. There are both positive and negative implications due to the almost

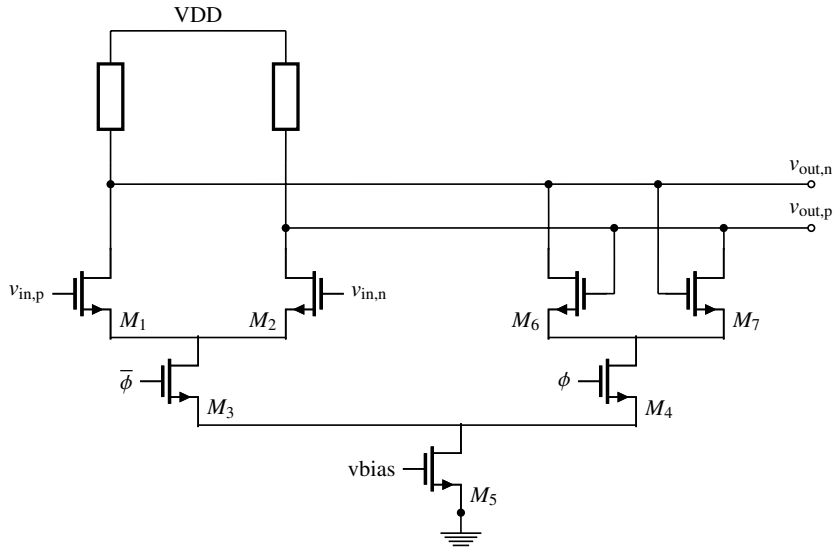

Figure 4.13: Current mode logic comparator schematic

static power consumption of this comparator implementation. On the one hand, no large transients are expected at the supplies which makes power decoupling easier. On the other hand, power does not decrease with operating frequency so long as the charging process of the clock transistors are not taken into account. However, for performance reasons, the supplied clock needs to be of full digital swing which always adds some variable component to the otherwise static power draw. The restoration to full level can for instance be achieved by a subsequent latch which then also takes care of retiming the signal. This latch then of course needs to be comparator based as well since it cannot be guaranteed that the CML low logic level will actually be smaller than the n-MOS device threshold potentially required by all-digital latch implementations (such as the PowerPC latch or D-latches for instance). However, due to the already higher voltage swings, the latch comparator has far less stringent demands associated with it than the first stage comparator and therefore has a smaller impact on overall sampler performance. Figure 4.15 depicts a possible sampler structure with a StrongArm based retiming latch as a second stage.

Strong Arm Figure 4.14 shows the evidently much more complex StrongArm comparator. Six of the thirteen transistors shown have the sole purpose of resetting the comparator to a well defined starting level at all nodes (reset/track phase). On the clocks rising edge, the input transistors start to pass a small signal difference seen at the inputs first to the nodes at their drain and finally into the cross coupled latch structure above by virtue of the small current difference in both branches caused by the difference in input transistor transconductance. This small initial bias is then amplified in the cross couple inverter structure. However, there is no small signal gain involved in this process due to the transient operating points of the involved transistors which at first are all in their triode region as the

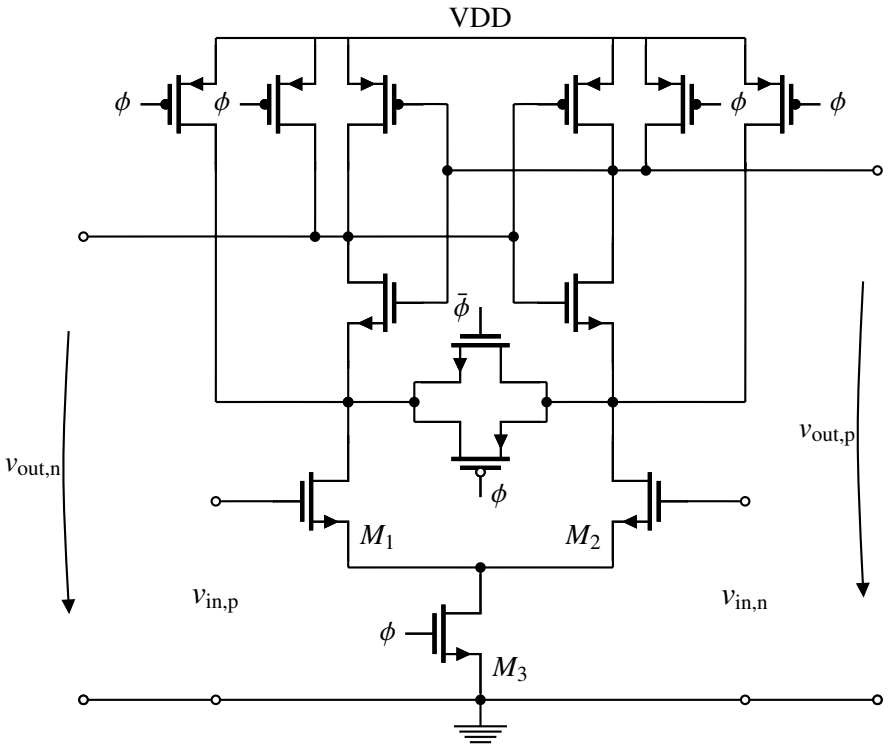

Figure 4.14: StrongARM comparator schematic

tail clock transistor starts to open.

As can easily be guessed, the clock transition produces quite large transients at the comparator supply which necessitates good power rail decoupling - even more so since the power supply noise rejection properties of this circuit are far worse than those of the CML implementation. Power consumption will scale with operating frequency. Especially when a serializer supports a range of data rates, this offers a potential power benefit. Furthermore, the circuit produces full digital output levels already by itself if given enough resolution time. Since the StrongArm comparator offers valid levels only at the end of the regeneration (hold) phase, it is usually supplemented by an unclocked RS-Latch. Retiming is then accomplished by a second latch stage of opposite clock edge sensitivity (see figure 4.16).

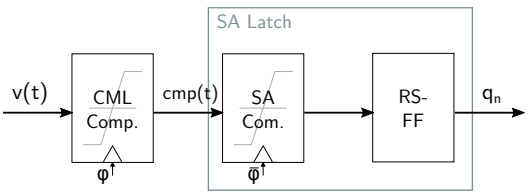

Figure 4.15: A possible implementation of a CML based sampler

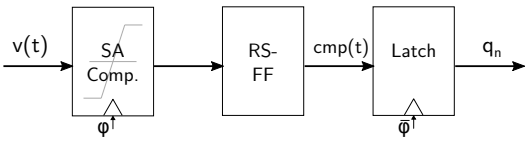

Figure 4.16: A possible implementation of a StrongArm based sampler

Despite of the architectural differences for sampler designs, a common modelling procedure is required for this essential building block. In contrast to the channel model of the previous section, however, there is no direct data model available for conversion to the real number model domain. A set of metrics needs to be found with which the impact of the sampler on overall system performance can be evaluated and budgeted efficiently.

Digital notions such as the setup and hold time metrics as well as the clock to output metric are of no value in this particular case, as the input voltage levels are purely analog. The essential question to be answered by the model in equivalence to the acutal circuit must therefore be: given a particular waveform $v(t - t_n)$ at the input of the comparator at and around sampling instant t_n, can the comparator resolve the potential difference at its inputs to a level detectable by the succeeding, digital latch within the given amount of regeneration time which is half the sampling clock period? To this end, this text proposes to use the sampler model as suggested in [62] as it both describes a model *and* an extraction procedure for this model that can be implemented for both schematic and real number based views. As a side effect, the resulting model will also allow to compare the CML and StrongArm implementation on a performance level.

4.5.1 Model theory

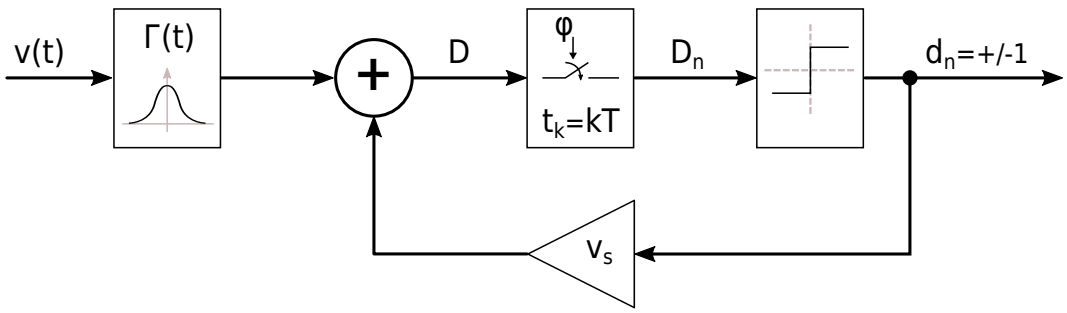

Figure 4.17: OCM sampler modelling methodology (after [62])

Figure 4.17 depicts the signal flow diagram of the sampler model conceived in [62]. It decomposes the sampler into an all analog, continuous and an all digital, periodic part. This is a consequence of the linear time variant (LTV) nature of the sampler system.

The analog representation is centered around a continuous filter structure (or weighting function) $\Gamma(t)$ which is called the impulse sensitivity function. It is defined with respect to and only within a single sampling clock period T such that $t = 0$ marks the sampling clock transition of $\phi(t)$, i.e. the instant at which the comparator changes from tracking to hold phase. The analog input signal to the comparator is continuously convoluted with the ISF to produce a continuous, analog intermediary decision value D. The ISF thus describes, how sensitive the sampler reacts to a voltage pulse of defined width t_{pw} at location t, i.e. what contribution is being made by the voltage magnitude $v(t - kT)$ to the final decision value D. This approach models the relative timing between sampling clock transition and the particular waveform $v(t)$ seen around this transition as a smaller voltage signal with rather weak slew rates would take much longer to arrive at a specific decision value than a large swing signal with fast transitions. Also, the relative timing between the rising clock of the sampler and the transition seen at the analog input is taken into account in this way. Start and end of the ISF can be conceived as some sort of optimal analog setup and hold times (albeit it is not garantueed the comparator will actually flip, since this also depends on the magnitude of $v(t)$). It is sensible to normalize the impulse sensitivity function (ISF) according to

$$\int_{\frac{-T}{2}}^{\frac{T}{2}} \Gamma(t)dt = 1 \tag{4.9}$$

since this allows to define the so-called *sampler sensitivity*. It governs how large the intermediary value D (and hence the magnitude of $v(t)$) will have to be in order to flip the sampler output. Due to potential residual charges in the sampler structures, the decision at time instant kT may actually depend on the previously sampled value at $(k - 1)T$. A sampler may either be more likely to resolve the same or opposite value to its previous decision which can be modelled by magnitude and sign of the sampler sensitivity v_{s}. This is reflected by the feedback amplification structure of figure 4.17. The intermediary decision value D hence equates to

$$D = \int_{\frac{(2k-1)T}{2}}^{\frac{(2k+1)T}{2}} v(\tau - kT)\Gamma(\tau)d\tau + v_{\mathrm{s}} \cdot d_{k-1} \tag{4.10}$$

From the ISF, two important performance metrics can be derived: The ideal sampler would make its decision instantly. Quite obviously, $\Gamma(t) = \delta(t)$ would be the appropriate

mathematical description in this case, or, in other words: the ISF is simply equivalent to a Dirac function. Since the ISF really models the impulse response of the time varying system, a frequency response and hence a bandwidth can also be associated with it. In the case of the ideal sampler, this bandwidth would of course be unlimited. Real and well-behaved samplers will, however, be represented by a peaking function of finite maximum, rolling off rather quickly as one moves away from the maximum (compare figure 4.22 below). This broadening of the Dirac function in turn leads to a drop in sampling bandwidth. The degree to which the peak broadens therefore represents a good measure of performance for a sampler and is called the *sampling aperture*.

In the literature, sampling aperture is often defined as the full width at half maximum of the peak. Here, we choose a more rigorous definition and demand that the level is not set at half of the peaks maximum but such, that eighty percent of the total area of the ISF is covered by the resulting integration interval symmetric around the ISF peak. The maximum of the ISF will not necessarily be located at $t = 0$ but can be offset by the so-called *aperture delay* which is just a consequence of the sampler device voltage thresholds, capacitances and the time required for the sampling clock signal to fully establish or impede the device channel conductances.

The digital part of the model consists of an ideal sample hold stage which produces D_n from D. Note, however, that this ideal sampling procedure does not hold the value of D instantly at $t_k = kT$ but needs to capture the convolution of ISF and $v(t)$ until the ISF magnitude becomes insignificant. As a result, the point in time at which the output of the comparator actually flips also strongly depends on the analog input voltage magnitude. This is in stark contrast to a digital latch, where clock to output time only changes mildly with the particular setup time (at least for setup time values decently distant from the metastable region). The second digital stage is an ideal decision stage which produces the sampler output according to

$$d_n = \begin{cases} +1, D_n \geq 0 \\ -1, D_n < 0 \end{cases} \tag{4.11}$$

The model described above can of course only be of use, if there is an efficient way to determine the ISF and sensitivity from actual implementations, i.e. if the model truly captures real circuit properties. To this end, different procedures have been devised both for simulation [62] or in modified form even for lab measurements [22]. As the ISF can, via Fourier transformation, be viewed in frequency domain, too, there is also a more advanced technique which uses periodic steady state (PSS) analysis in conjunction with Periodic AC Analysis [27]. Mathematically, this procedure is more difficult to handle but simulates magnitudes faster. It is however not suitable for real number models since circuits to be analzed by PSS may not have *hidden states* which all kinds of registered values as they

usually appear in SystemVerilog code certainly are.

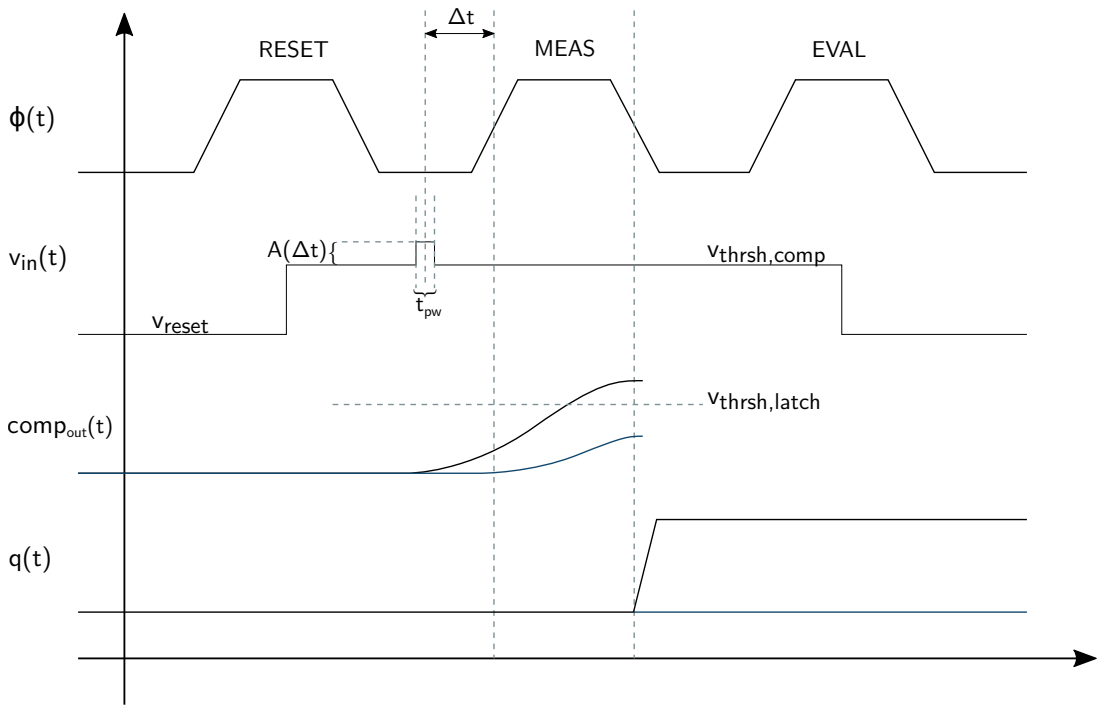

Figure 4.18: OCM sampler ISF measurement methodology (after [62])

Figure 4.18 highlights the atomic stimulus sequence to capture the required information to extract a samplers ISF and sensitivity according to [62] with some variations specific to the implementation developed and presented here. It assumes a sampler sensitive to the rising edge of the clock. The sequence is atomic in the sense that it needs to be repeated for all time offsets Δt between sampling clock edge and pulse center within the clock period. It also needs to be repeated for a single setting of Δt since it must find the minimal pulse amplitude $A(\Delta t)$ which still leads to a detectable bit flip at the sampler latch output. It does so by using a *binary search algorithm* to converge to a good estimation of this value.

The atomic sequence itself goes about as follows: First, the sampler is *reset* with a large overdrive voltage v_{reset}. The applied input voltage is then set to the assumed comparator threshold value (i.e. zero differential voltage for the circuits as given above). At some point within the *measurement* period, the pulse of amplitude A is applied at the required offset Δt to the sampling edge, the signal remains at threshold value elsewhere. The falling edge of the measurement phase marks the point at which the latch subsequent to the comparator needs to be able to capture the resolved digital value which can then safely be sampled at the rising edge of the *evaluation* phase. This is in contrast to the procedure defined in [62] where the comparator was probed for an explicit, analog output voltage magnitude set to the mid-rail voltage. Here, we do not particularly care about the actual absolute

resolution voltage of the comparator but conceive the subsequent retiming latch as part of the sampling process. This is appropriate considering the fact that the input capacitance of the retiming latch actually loads the regeneration structure of the comparator thus reducing its time constant. The output voltage of the regeneration latch is then compared against the standard logic threshold values of the node technology at hand. Figure 4.18 depicts the situation for two different sampler instances (black and blue curves for $comp(t)$ and $q(t)$) only one of which successfully resolves the pulse of distinct height and temporal displacement in the given situation.

It has been shown in [62] why and how this algorithm finds the ISF and sensitivity value of a sampler. The arguments are repeated here in slightly modified notation:
The algorithm determines $A(\Delta t)$ such that

$$A(\Delta t) \cdot \int \Gamma(t)\mathrm{rect}(\Delta t - \tau, t_{\mathrm{pw}})d\tau = v_{\mathrm{s}} \tag{4.12}$$

holds. The pulse width t_{pw} used in the simulation will set the resulting temporal ISF resolution. Equation 4.12 is equivalent to writing

$$\frac{v_{\mathrm{s}}}{A(\Delta t)} = \Gamma(t) * \mathrm{rect}(t - \Delta t, t_{\mathrm{pw}}) \tag{4.13}$$

and with $t_{\mathrm{pw}} \to 0$ it follows that

$$\frac{v_{\mathrm{s}}}{A(\Delta t)} \approx t_{\mathrm{pw}}\Gamma(\Delta t) \tag{4.14}$$

Both $A(\Delta t)$ and $\Gamma(\Delta t)$ are discrete functions in time due to the derivation procedure. For ease of writing, we let $\Delta t \to t$ here for the moment. The ISF can then be written as

$$\Gamma(t) \approx \frac{v_{\mathrm{s}}}{A(t)} \tag{4.15}$$

from which the sampler aperture can be computed by finding the area symmetrically around the peak value which covers eighty percent of the total area. The sampler sensitivity can be found from the normalization condition of the ISF:

$$\int_{\frac{-T}{2}}^{\frac{T}{2}} \frac{v_{\mathrm{s}}}{A(t)}dt \equiv 1 \tag{4.16}$$

from which the sensitivity can be calculated to

$$v_{\mathrm{s}} = \frac{1}{\int \frac{1}{A(t)}dt} \tag{4.17}$$

The measurement procedure is very sensitive to right proportions of pulse width, sample clock rise and fall times and simulator accuracy settings. Also, with very good signal rise and fall times, a high ISF resolution is of the essence.

In figures 4.19 through 4.21 the resulting ISF and frequency responses of two different StrongArm sampler implementations in a 28 nm technology are shown. The comparator dimensions are given in table 4.8. The extracted sampler metrics are listed in table 4.9 for rise times of 25 ps and a sampling frequency of 10 GHz (and hence a resolution time of 50 ps). For the first sampler version the ISF is also measured with a variation in sampling period T and clock slew rate.

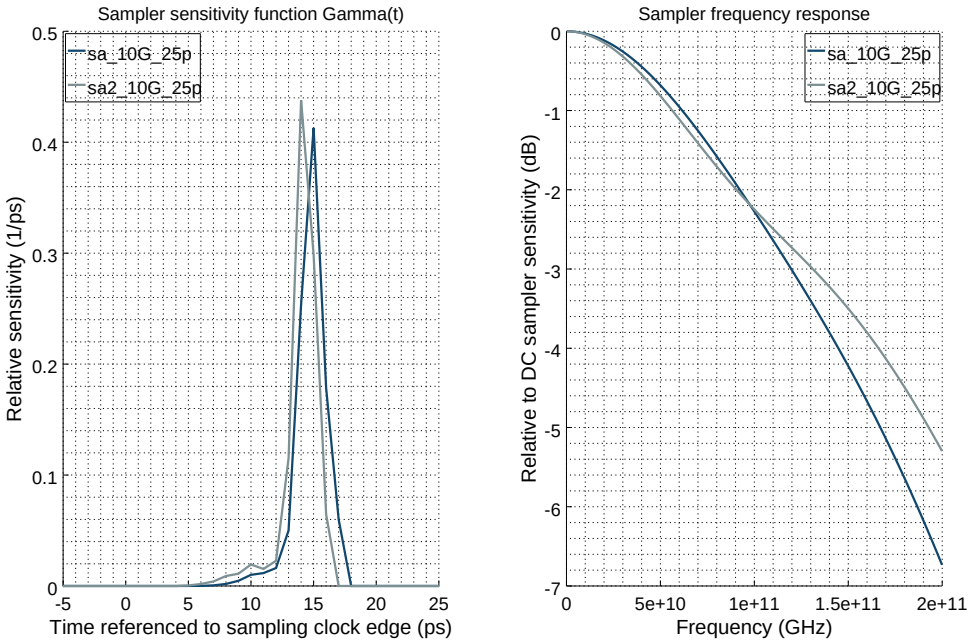

Figure 4.19: ISFs and frequency responses of the two sampler versions with dimensions as given in table 4.8

Comparing the two ISFs in figure 4.20 we can see that the shape of the ISF is a property of the sampler structure and does not change (much) with sampling clock periodicity T.

Sensitivity on the other hand is a property of T. While the sampler under test exhibits a sensitivity of 1.6 uV at 5 GHz and 25 ps clock transition time, this value increases to 11 uV at 10 GHz sampling frequency. A longer resolution time will allow smaller input signals to still be resolved correctly.

The ISF is very sensitive to changes in sampling clock rise and fall times (as of course is the aperture delay, see figure 4.21). At 10 GHz sampling frequency and a clock rise/fall time of 25 ps, the aperture delay is 15 ps and the aperture width is 2.8 ps resulting in a sampling bandwidth of 119 GHz. The same sampler supplied with longer clock transition times

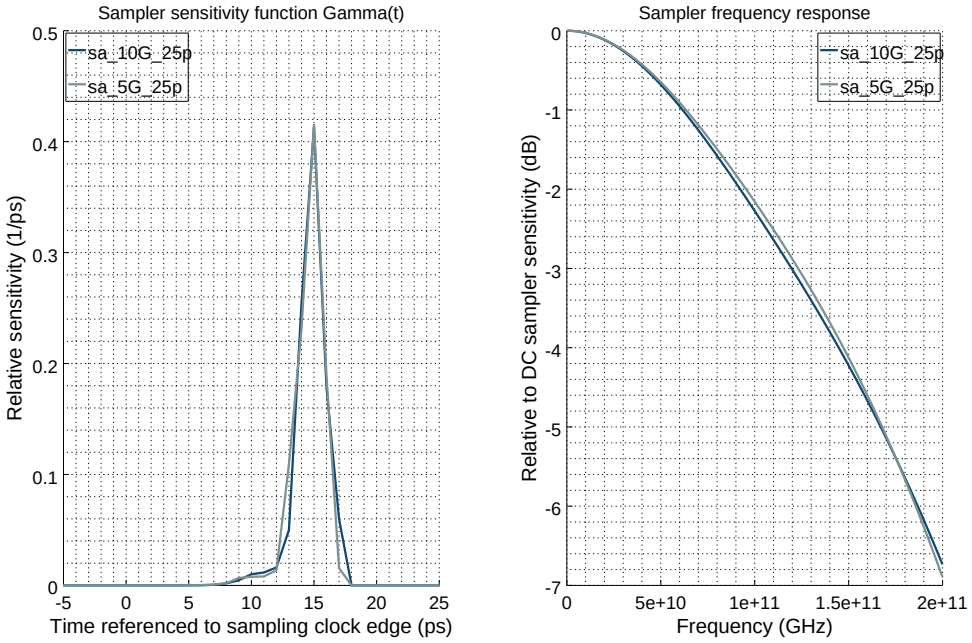

Figure 4.20: ISFs and frequency responses for the StrongArm sampler with comparator dimensions as given in the first row of table 4.8 for two different sample clock frequencies (and therefore periods T)

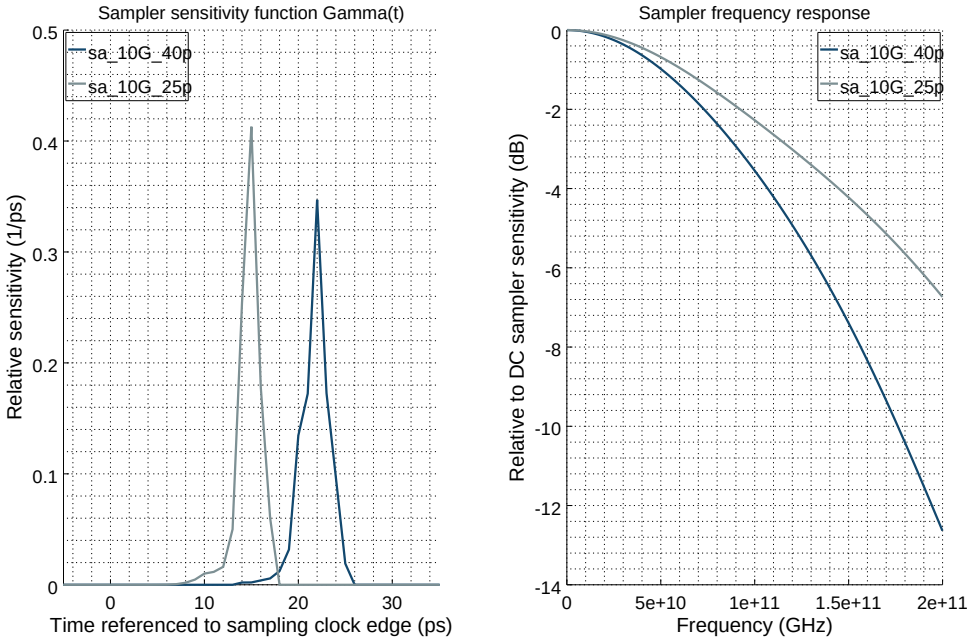

Figure 4.21: ISFs and frequency responses for the StrongArm sampler with comparator dimensions as given in the first row of table 4.8 for two different sample clock slew rates t_{rf}

of only 40 ps leads to an increase in aperture delay to about 22 ps, increases the aperture width to 3.8 ps (which decreases the bandwidth to 91 GHz) and drastically decreases the

sensitivity to 200 uV.

Version	Input M_1, M_2	Tail Clock	Cross Coupled FB M_{n-m}	VDD Reset M_{n-m}	Shunt Reset M_x, M_y
1	7 um	10 um	1.6 um	1.8 um	0.2 um
2	3.5 um	7 um	0.4 um	0.42 um	0.12 um

Table 4.8: Gate widths of the transistors shown in figure 4.14 for the two exemplary versions of a StrongArm comparator discussed here.

Version	Sensitivity	Aperture width	Aperture delay	Bandwidth
1	11 uV	2.8 ps	15 ps	119.7 GHz
2	46 uV	2.4 ps	14.3 ps	131.2 GHz
CML	27 uV	8.2 ps	7.8 ps	41.9 GHz

Table 4.9: Performance parameters for the two StrongArm sampler versions of table 4.8 and a CML realization

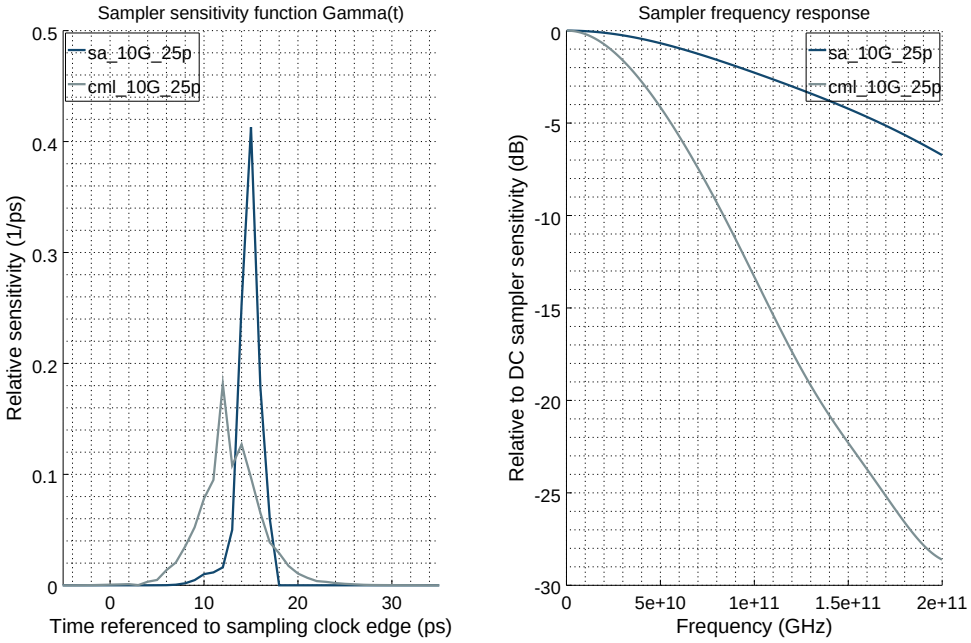

Figure 4.22: ISFs and frequency responses for the StrongArm sampler with comparator dimensions as given in the first row of table 4.8 and a CML sampler of the same input transistor size

Figure 4.22 and table 4.9 compare the first StrongArm version of table 4.8 to a CML implementation with equally sized comparator input transistors. As can be seen from the

graphs, the aperture width of the CML implementation is much wider which leads to a drastic decrease in sampling bandwidth. The sensitivity on the other hand maintains a very good level of 27 uV and its aperture delay is significantly lower compared to the StrongArm version.

Therefore, in addition to the advantages and disadvantages of the two implementations as described above, the measurements indicate the very small aperture width (and therefore high bandwidth) and good sensitivity as further strong points of StrongArm based samplers compared to their CML counterparts whereas CML versions have an additional benefit in providing small aperture delays.

As one can see from figure 4.19 almost the same ISF shape and therefore aperture width can be realized with smaller or larger devices within the comparator. Why then would a designer still opt for the larger version of a sampler? It seems more benefitial to use very small devices with fast clock slew rates if the sampler sensitivity is of much concern. After all, a large device will certainly produce more clock load and higher transient currents! One reason is that device mismatches drastically increase with smaller device sizes and more static offset correction would then be needed - an effect that becomes more severe with more advanced technology nodes. Perhaps the more important answer however lies in the intrinsic noise properties of the sampler.

Intrinsic noise analysis First, it is rather difficult to find a good estimate of sampler noise from analytic calculations. Especially in the case of a StrongArm comparator there is a substantial amount of device noise sources to be considered. Next, for a very nonlinear device such as a sampler or latch, there is no direct way to refer the noise seen at its full rail outputs back to a representative noise magnitude at its inputs - the so called *input referred noise*. However, for serializer link budgeting we ultimately require a metric value which allows to calculate the impact of sampler intrinsic noise on the estimated eye diagram seen by the receivers sampler inputs. Furthermore, a good portion of the overall noise level will result from periodic and thus deterministic effects, most notably the clock feedthrough. This is also called *cyclostationary noise* and cannot be covered by small signal models.

There has been a procedure suggested in [54] and extended by [57] to find the input referred noise $\sigma_{n,samp}$ of a sampler. It is based on a transient simulation with all intrinsic noise sources of all components within the sampler activated. Since the sampler design is rather small, a single simulation run is still fast enough to cover several hundreds of nanoseconds. This can be done even when choosing the noise bandwidth of the transient, noise enhanced simulation to be rather large. As the lower bandwidth limit also sets a limit on minimum simulation time, this value needs to be chosen with respect to run time rather than by physical indication, i.e. in the range of some MHz. This will then of course not capture the steeply rising $1/f$ noise components of the devices. However, since the upper bound will be chosen in accordance with the samplers bandwidth as obtained from

ISF extraction (and therefore in the vicinity of 100 GHz), the $1/f$ noise which is higher in magnitude than thermal noise contributions but much smaller in bandwidth will be overwhelmed by noise power integration over the huge remainder of inband thermal noise.

The testbench applies a small DC offset V_{off} to the sampler input and records the number of logic zeros and ones at the output of the sampler. The input therefore needs to be chosen small enough to still allow intrinsic noise to alter the comparators decision. Due to the mostly thermal noise contribution and due to the central limit theorem of statistics, the noise PDF is expected to be of Gaussian nature. The number of detected ones n_0 and zeros n_1 are therefore expected to be distributed according to [54].

$$\frac{n_0}{n_1} = \frac{\text{erf}\left(\frac{-V_{\text{off}}}{\sqrt{2}\sigma_{\text{n,samp}}}\right)}{\text{erfc}\left(\frac{-V_{\text{off}}}{\sqrt{2}\sigma_{\text{n,samp}}}\right)}$$

This equation can be solved numerically for $\sigma_{n,samp}$. However, the above formulation neglects the fact that the overall Gaussian process actually possesses a second, free parameter next to the standard deviation - the mean value μ, which in this case represents an intrinsic sampler offset. It comes about due to the potentially very slightly nonsymmetrical detection thresholds of the latch succeeding the comparator. Small modifications in μ will lead to much different results for the estimated σ in the procedure above. Also, the process described thus far suffers from statistical fluctuations if the DC offset magnitude is chosen in too much proximity to the mean value of the Gaussian distribution.

Therefore [57] extends the idea by performing a set of transient simulations with various values for the DC offset. The sweep allows to derive a cumulative density function (CDF) in dependence on V_{off} where each point of the CDF is computed as

$$\text{cdf}(V_{\text{off}}) = \frac{\frac{n_1 - n_0}{n_0 + n_1} + 1}{2}$$

Its derivate then is the PDF of the noise process and the free parameters of the Gaussian PDF of equation 2.6 (see section 2.1) can be fit to the measurement. This procedure achieves a higher accuracy and also finds the static intrinsic offset (if present).

As can be seen from figures 4.23 and 4.24, wider devices within the sampler lead to a reduction of the sampler intrinsic noise level. This is due to the increase in capacitance at the internal nodes of the comparator. The slope of the CDF is much steeper for the first dimensioning effort compared to the second and as a result, with the PDF being the derivative of the CDF, the standard deviation of the underlying Gaussian process must consequently be much smaller. Also note that once the noise level becomes comparable to the static sampler input offset used in the testbench, the derivative of the CDF becomes increasingly noisy too. Towards the tails of the Gaussian function in figure 4.24, this effect

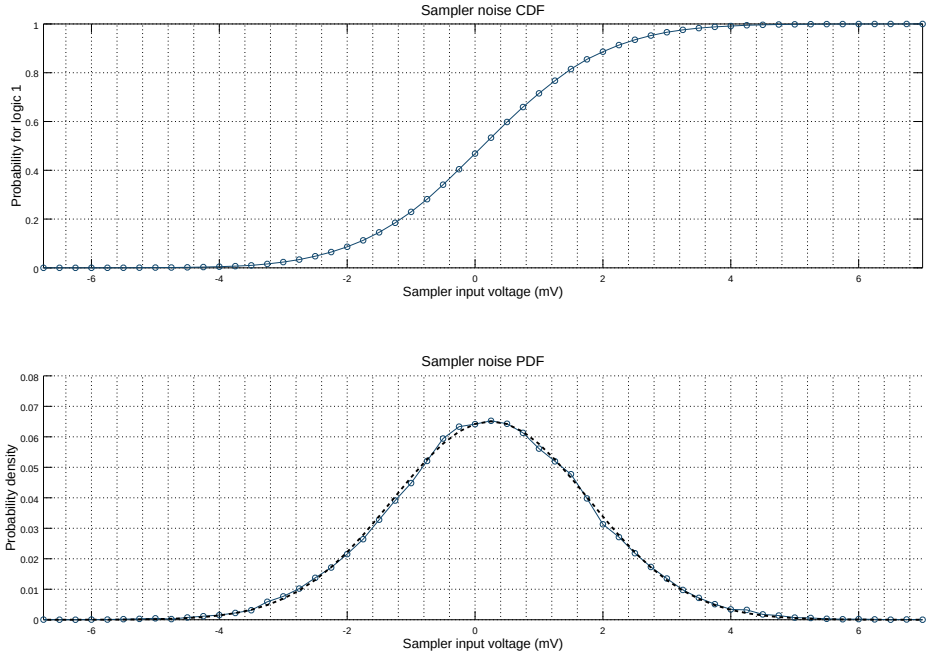

Figure 4.23: CDF and PDF of a strongARM comparator with dimensioning according to first row in table 4.8

can be seen to decrease as expected. Also, table 4.10 lists the two free parameters of the fitting process (inidicated by the black dashed lines in the PDF subfigures), the standard deviation σ and the mean μ, i.e. the sampler intrinsic offset. It is this initially unknown offset along with the equally unspecified magnitude of the input refferred, intrinisc noise of the sampler which renders the attempt of deriving the intrinisc noise at solely a single static input value an unreliable effort.

For completeness, figure 4.25 shows the noise analysis result for the CML based sampler. As can be seen, its input referred noise is much lower compared to the StrongArm implementations. Partially, this is owed to the much smaller bandwidth of the sampler itself. The level of noise for the devices presented here quite generally show that in lab measurements, it should be rather difficult to extract the theoretical DC sensitivity of a sampler as its value may be magnitudes lower than the intrinsic noise level.

Version	**Input referred RMS noise** $\sigma_{n,samp}$	**Intrinsic offset** μ
1	1.53 mV	250 uV
2	4.27 mV	14.7 uV
CML	0.87 mV	93 uV

Table 4.10: Noise parameters for the two StrongArm sampler versions of table 4.8 and a CML realization

In conclusion the aperture width, its delay, the sampler sensitivity, the intrinsic offset as well as the intrinsic noise of the sampler are a good set of metrics to define sampler performance. However, during design space exploration, there is initially no schematic level implementation available from which the ISF and the resulting parameters could be drawn. Therefore, for the RNM modelling procedure we also require an abstract function which captures the general trend of an ISF and allows to abstractly model the sampler. System level budgeting may then be used to arrive at acceptable upper bounds to the sampler metrics.
This text suggests the *Cauchy-Wigner* function for this type of abstract ISF as it provides the necessary steep rolloff required to model StrongArm based comparators. CML based comparators on the other hand may be well approximated by a Gaussian bell function. Both must of course be truncated to a maximum width. The Cauchy-Wigner function is defined as

$$f(x) = \frac{1}{\pi} \frac{s/2}{\left(\frac{s}{2}\right)^2 + (x - x_0)^2} \tag{4.18}$$

which implicitly guarantees the normalization condition. While x_0 can easily be seen to represent the aperture delay, s really represents the full width at half maximum of the function. In order to comply with the definition of the sampling aperture as declared above, we need to find a relation between s and the definition. To this end, since the antiderivative of $f(x)$ is known to be

$$F(x) = \frac{1}{\pi} \arctan\left(\frac{2(x - x_0)}{s}\right)$$

the relation

$$n = \int_{x_0 - a/2}^{x_0 + a/2} f(x)dx = \frac{2}{\pi} \arctan\left(\frac{a}{s}\right)$$

must hold where n is the fraction of area to be covered by the integration (which due to the aperture definition is set to 0.8). With the exception of the ISF shape only, this in conjunction with SystemVerilog and C code presented below allows an implementation agnostic model definition which captures the relevant parameters of the sampler.

A precise yet computationally efficient model for the sampler is absolutely essential to serializer design space exploration and budgeting as its metrics will have repercussions on

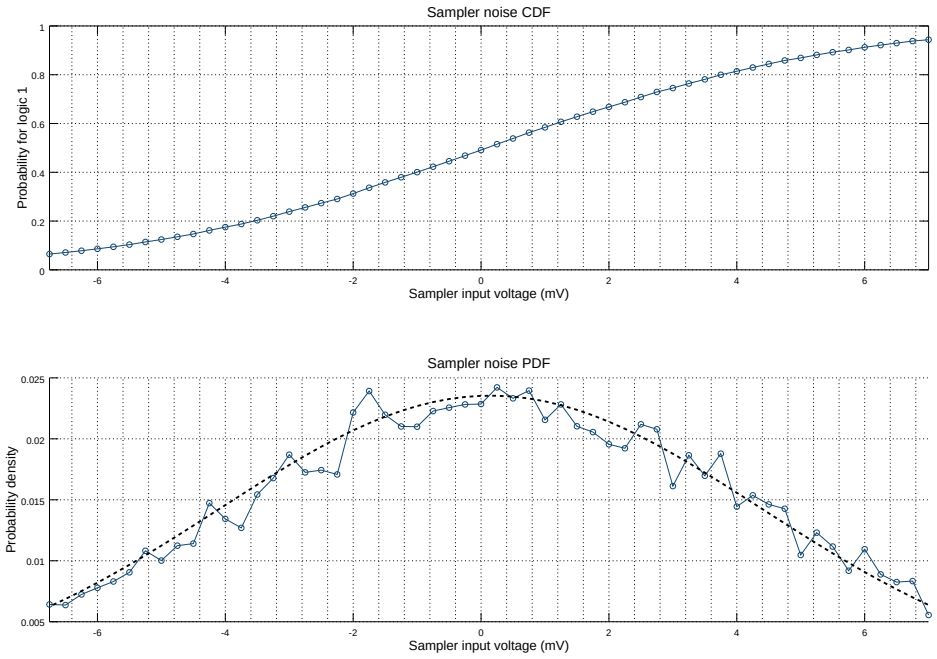

Figure 4.24: CDF and PDF of a strongARM comparator with dimensioning according to second row in table 4.8

design parameters throughout the receiver:
For the analog realm, the combined input capacitances of all samplers present the load to the analog frontend amplifier chain and therefore set the amount of power consumed there. On the other hand the noise figure of the amplification chain benefits from greater capacitances at the sampler input and in between the stages of the amplifier chain as does the self noise of the sampler - the well known power/SNR tradeoff. The overall noise level at the sampler input in turn determines the analog frontend gain to allow equalization and operation of a particular physical channel at the specifed BER target.

For the digital realm, the samplers may also set the upper limit of operation frequency once the resolution time of the comparator exceeds the UI. More importantly though, the ISF of the sampler will directly impact the phase detector gain K_{PD} of the clock data recovery loop. It will also affect the timing budget for the first tap of the direct feedback equalization loop and may thus force a speculative implementation approach (see chapter 3).

4.5.2 Models / OCM Data Sources

The general model theory now allows to develop an efficient implementation of a real number sampler model. The numeric backend of OCM supports three distinct model

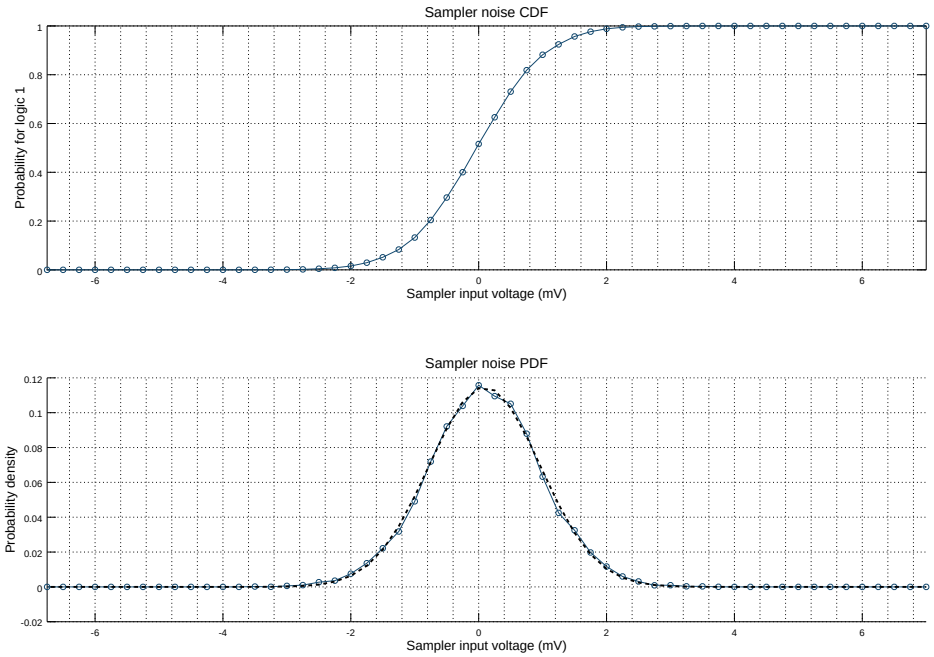

Figure 4.25: CDF and PDF of a CML comparator of same input transistor size as the StrongArm sampler with comparator dimensions as given in the first row of table 4.8

definitions. The most simple version is the abstract Dirac sampler as a default fallback solution. For budgeting, a parameterizable Cauchy-Wigner model as presented above can be used. A third option lies in using the ISF of a schematic implementation (i.e. for final budgeting) as recovered from the testbench whose structure is also shown in figure 4.26. It must be noted, however, that extracted models are only accurate for the sampling clock period and clock signal slew rates they were extracted with. While the clock slew rate is fixed by design, the sampling clock period may be changing for serializer systems which support multiple rates of operation. It is then advisable to recover the ISF by using the shortest possible period to minimize resolution time and thus the sensitivity estimation. It also maximizes the noise properties which is advisable for conservative budgeting.

The numeric backend again provides the benefit of keeping the sampler model in synchronization between the transient simulations and the numeric post process treatment such as budgeting. Especially for the so-called post aperture eye (see section 5.3) and the derivation of the phase detector gain in the CDR loop, this is of much benefit. Much like the channel, the sampler model can be dynamically adjusted with respect to simulation resolution. Due to the very small aperture widths of samplers in advanced technology nodes, however, the model ISFs quickly converge towards a Dirac like implementation.

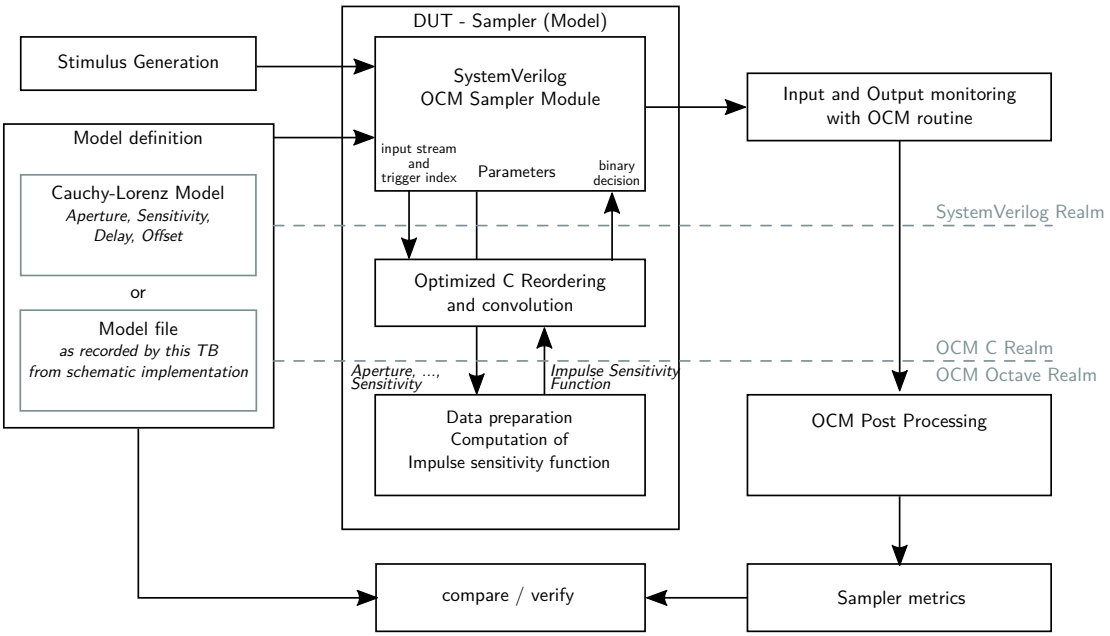

Figure 4.26: OCM sampler modelling and testbench flow

4.5.3 SystemVerilog considerations

The SystemVerilog module of the sampler continuously records the stream of real number values which represent the voltage difference seen at the sampler input. The depth of the history it records spans an adjustable integer multiple of the sampler aperture width commonly set to three. Generally, it would be possible to also continuously convolute the input data with the FIR-like filter defined by the ISF. However, in contrast to the channel module, there may be quite a number of sampler instances in a given realization of a receiver. The design presented in chapter 6 requires a total of fourteen sampler instances. Continuous convolution in all these instances in order to obtain just a single value per period would certainly consume a lot of computing resource without any apparent benefit. Therefore, the sampling module calls its DPI convolution function only once when triggered by its clock input. Since the ISF may span to either side of the trigger instant, the history will be recorded even after the trigger event occured until the specified integer multiple of the ISF aperture is fully recorded. Even though the convolution function returns with its binary decision within the same time slot of the simulator, this introduces additional delay in the clock to output characteristic of the comparator. Since the sampler structure always consists of a comparator element whose output is retimed by at least a single latch stage, this does not pose a problem if the post trigger history item count does not exceed the resolution time of half the sampling period.

4.5.4 Optimized C modelling routine

Since the input clock to the sampler may be perturbed by duty cycle distortion or jitter, there is no known static value for the ring buffer pointer of input data history acquisition for every trigger event. The input history, however, needs to be convolved with the ISF under alignment of sampling instant, as recorded by the SystemVerilog module, and the ISF data item which represents $\Gamma(0)$. Therefore, the buffer pointer (or *trigger index*) is copied to the C domain as well, where the history buffer is rearranged with efficient memory copy operations which then allows direct convolution with the model ISF. The SystemVerilog DPI call for sampler value resolution is thus defined in the following way.

```
import "DPI-C" function bit ocmSamplerGetResult (
    input int cid,
    input real historyArr[],
    input int ti
);
```

The sampler id *cid* is used to support multiple sampler instances with potentially different model characterisitcs. It directly returns a C representation of a simulator bit value requiring no further conversion or compare efforts within SystemVerilog.

Again, the convolution routine is optimized by using SIMD instruction set extensions. There is, however, no real benefit from spawning a thread per sampler as control to the sampler resolution routine is passed from the single threaded simulator to only one sampler instance at a time. The C code also takes care of initialization, reads back the resulting sampler metrics and the ISF from the numeric OCM backend into internal data structures to correctly allocate memory and reflect model operation.

4.5.5 Testbench and performance analysis

The sampler testbench probes the sampler for the ISF according to the procedure described above. It is written such that it supports both running in conjunction with schematic implementation or a real number model instance. In order to speed up the characterization process, the testbench uses an adjustable amount of identical sampler instances and stimulus generators in parallel to alleviate the single threaded nature of the digital simulation kernel. The analog kernel can then make use of its multithreaded optimizations. This reduces the overal simulation time for a completely schematic based instance from roughly one hour for single core execution to just about nine minutes and 4 seconds with high accuracy presets and twenty instances in parallel. The run time is of course dependent on sampling clock period and actual sampler sensitivity - more sensitive implementations force the binary search algorithm into probing more values.

Figure 4.27 shows the results for an ISF extraction run of a StrongArm sampler schematic implementation together with its file based real number model. As can be seen, the recovered ISF and metrics are truly identical. This also proofs the model and testing procedure of

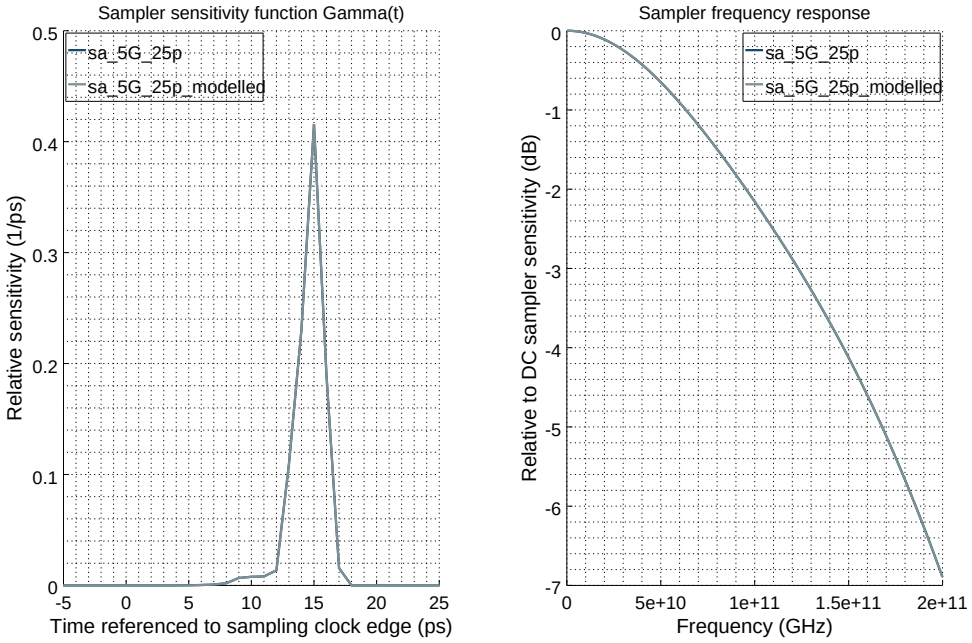

Figure 4.27: ISFs and frequency responses for the StrongArm sampler with comparator dimensions as given in the first row of table 4.8 and for its file based real number model.

the sampler to be *mutually consistent*. The time required for running the real number model based simulation was **4.1** s with again a total of twenty sampler and stimulus instances used in parallel. This results in a speed up on the order of **132** compared to the multithreaded execution and a factor of **900** for single core execution of the analog simulation kernel.

With the most computation demanding serializer system components so modelled, the next chapter turns its attention to the overall system budgeting and analysis procedure and to the question, how transient simulation time can be decreased even further, especially to capture the deterministic effect of intersymbol interference caused by high loss transmission lines.

5 Link budgeting

Since the advent of multi-gigabit, baseband communication channels about two decades ago, several techniques have been developed to assist in the design of these systems and their constituent subcomponents. As discussed in chapters 2 and 4, each subsystem has a set of performance metrics. They may each have a more or less subtle impact on one or more toplevel system metrics, either directly or by interaction with other subcomponent properties. An obvious example of this may be the voltage noise spectral density $S_{vv,\mathrm{pdn}}$ of the PDN which impacts the noise level seen at the receiver samplers via the various power supply rejection ratios of the receiver amplifier stages. It is a simple example of how one subcomponent metric cannot be utilized or constrained in a meaningful manner without knowing and constraining another subcomponents metric.

It is the central task of the link budgeting procedure to establish the relations between all important subcomponent metrics (if they exist) and their combined or isolated impact on key system performance metrics. For the above mentioned example, the affected toplevel metric of $S_{vv,\mathrm{pdn}}$ and the PSRR would of course be the eye height or the expected BER of the system itself. Once these relations are established, budgeting helps to identify the strongest contributors to system performance degradation. Some of these metrics, however, may be subject to other constraints such as power consumption in which case other, related metrics have to be improved in its stead.

The goal here is to develop an approach for the openMGT/OCM framework which allows to budget a system as described in chapter 2 without resorting to third party tools and parameter synchronization. To that end, the algorithms found in the papers presented here will be adapted and modified as needed. In order to make this process comprehensible, the ideas and steps taken from the papers will be repeated here albeit with adapted nomenclature and mathematical convention for improved consistency, especially with the peak distortion analysis (PDA) algorithm of section 5.2. The starting point of almost every approach is the physical transmission channel itself as its impact on required system components and their metrics is by far the strongest. The influence of voltage noise sources such as the power supply noise and the thermal noise of the circuit elements is well understood and can be treated independently of the transmitted signal. This is due to the linear time invariance of the TX FIR filter, the channel and the RX CTLE structure. Special care must only be taken with the CDR circuit (see section 5.3.6) as voltage noise will impact key design parameters of this subcomponent and cause nonlinear backaction

on the subcomponents behavior. The central question of budgeting therefore is, how jitter sources in the system are being dealt with in the presence of high loss transmission lines and linear equalization efforts.

5.1 History and state of the art

Let $H(j\omega)$ be the frequency transfer function of the transmission channel terminated at transmitter and receiver. Its impulse response $h(t)$ can be found as usual by Fourier transformation. The channel response to an arbitrary input waveform $x(t)$ can then be computed by convolution to

$$y(t) = h(t) * x(t) = \int_{-\infty}^{\infty} h(t)x(t - d\tau)d\tau$$

Further, let $\varepsilon(t)$ represent the instantaneous time interval error of $x(t)$ (see section 2.2.1). This of course only makes sense, if $x(t)$ has a fundamental periodicity T. As $x(t)$ is the continuous time representation of a stream of digital bits, this periodicity T is given by the unit interval of a bit. As mentioned in section 2.1, $x(t)$ can be rewritten as

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega$$

and thus

$$x(\ t + \varepsilon(t)\ ) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(X(\omega)e^{j\omega\varepsilon(t)}\right) e^{j\omega t}d\omega$$

As expected, the Fourier transform would lead to a completely new set of coefficients $\overline{X(\omega)} = X(\omega)e^{j\omega\varepsilon(t)}$. However, if the condition $|\varepsilon(t)| \ll T$ holds, i.e. the time interval errors remain small compared to the unit bit interval, $\overline{X(\omega)}$ can be Taylor expanded:

$$\begin{aligned} x(t + \varepsilon(t)) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(X(\omega)e^{j\omega\varepsilon(t)}\right) e^{j\omega t}d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} (X(\omega)e^{j\omega t}\left(1 + j\omega\varepsilon(t) - \frac{\omega^2\varepsilon^2(t)}{2} + \frac{j\omega^3\varepsilon^3(t)}{6} + ...\right)d\omega \\ &= x(t) + \varepsilon(t)\frac{dx(t)}{dt} + \frac{\varepsilon^2(t)}{2}\frac{d^2x(t)}{dt^2} + ... \end{aligned} \tag{5.1}$$

As a shorthand writing, we will define the second term $j_1(t) = \varepsilon(t)\frac{dx(t)}{dt}$ as the jitter term of first order. For the small perturbations of the transmitted signals discussed here, the timing error $\varepsilon(t)$ is converted to a voltage error $j_1(t)$ via the slew rate of the unperturbed signal $\frac{dx(t)}{dt}$.

It is interesting to note that equation 5.1 is also encountered in the analysis of noise power spectral densities and therefore timing errors in oscillating systems such as PLLs. The output of an oscillatory system with timing error can be written as

$$c(t) = Asin(\omega_c(t + \varepsilon(t)) \approx Asin(\omega_c t) + A\phi(t)cos(\omega_c t)$$

where $\frac{dc(t)}{dt} = \omega_c cos(\omega_c t)$ and the phase error $\phi(t) = w_c\varepsilon(t)$ with ω_c being the carrier frequency. The power spectral density of $\phi(t)$ then is the noise power spectral density $S_{\phi\phi}$ of the system from which the RMS jitter is usually calculated.

Before arriving at the time discrete description of link budgeting as suggested in [60, 61] and further developed in [44], a swift look at the continuous point of view is given here first:

The sequence of time discrete input bits a_k is converted to an analog, time discrete transmitter output amplitude b_k via the filter matrix $\overline{\boldsymbol{W}}$ which is an N_w by N_w matrix where N_w is the number of filter taps (see figure 3.11 in section 3.3.1). Each bit is transmitted as an ideal rectangular pulse $p(t)$ with height b_k. The time continuous pulse function is given by

$$p(t) = rect\left(\frac{t}{T}\right) = \begin{cases} 1, & -\frac{T}{2} < t \leq \frac{T}{2} \\ 0, & \text{else} \end{cases}$$

$p(t)$ is not continuous and therefore not differentiable in the classical sense. Its time derivative can however be discussed in terms of the Dirac distribution as is then given by

$$\frac{dp(t - nT)}{dt} = \delta(t - nT) - \delta\,(t - (\text{n+1})T)$$

The input sequence along with timing error and expanded to first order then reads

$$x(t) = \sum_{n=-\infty}^{\infty} b_n p(t - nT) + \sum_{n=-\infty}^{\infty} b_n\varepsilon(t)\,(\,\delta(t - nT) - \delta(t - (\text{n+1})T)\,) \tag{5.2}$$

An expression for the output waveform of the channel with jitter is obtained by convolving the above expression with the channel response. This gives meaning to the distribution functions in equation 5.2 and due to linear independence of the summation terms and

convolution leads to

$$y(t) = \sum_{n=-M}^{N} b_n s(t - nT) + \sum_{n=-M'}^{N'} b_n \left(h(nT)\varepsilon(nT) - h((\mathrm{n}+1)T)\varepsilon((\mathrm{n}+1)T) \right) \tag{5.3}$$

where $s(t) = h(t) * p(t)$ in the first term of the right hand side is the well-known channel SBR. In the summation above, it is assumed that its maximum can be found at $n = 0$ and that it is limited to the range of significant SBR taps prior to ($-M$, i.e. precursors) and after the main tap (N, i.e. postcursors, see also figure 5.5 in section 5.2). The second term in equation 5.3 represents the waveform perturbation due to the time interval error (i.e. jitter) to first order. As mentioned above, it maps the timing error given by $\varepsilon(t)$ to an equivalent voltage noise (EVN) via the impulse response of the channel or system. Here, this can be seen more clearly by noting that the single bit response $s(t)$ can be decomposed into two step responses $r(t)$ with rising and falling edge $s(t - nT) = r(t - nT) - r(t - (n + 1)T)$. The step responses can be related to the impulse response of the channel via $h(t) = \frac{dr}{dt}$. Thus $h(0)$ is the rise time (slope) of a lone step at the zero crossing point (where the signal interpreted as NRZ changes sign). The summation over all significant taps of the impulse response ($-M'$ to N') then includes the bit pattern and signal level dependent effects on the jitter seen at the output of the channel or system.

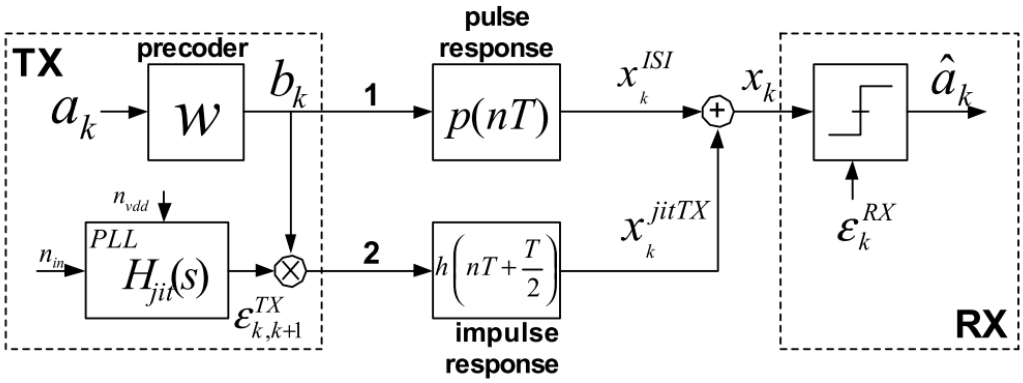

Figure 5.1: System overview for budgeting according to Stojanovic et al. [60]

The central ansatz of Stojanovic et al. [60] is a time discrete representation of the system response to a sequence of binary input symbols where the transmitter driver and its FIR filter act as a digital to analog converter (DAC). The LTI system of channel and FIR and its response to a sequence of binary input symbols denoted by $\boldsymbol{a}$, where $a_j = a(j \cdot T) = \pm 1$

can be written as

$$y_{\text{total}}(t) = \sum_{j=-M}^{N} b_j s(\varepsilon_{\text{tx},j} - jT)$$

where again the binary input symbols a_j are encoded to analog output amplitudes $b_j = b(j \cdot T)$ via the coding matrix $\overline{\boldsymbol{W}}$. The output driver retimes the transmitted symbols to its local PLL reference clock which has a timing error function $\varepsilon_{\text{tx}}(t)$ and where the transition to the time discrete description is again made by setting $\varepsilon_{\text{tx,j}} = \varepsilon_{\text{tx}}(jT)$. There is no timing error accumulation taking place in the transmitter, therefore only the particular timing error $\varepsilon_{\text{tx},j}$ at the nominal time instant jT is important here. At time instant $t = mT + \varepsilon_{\text{rx},m}$ the receiver samples the incoming data with its jittered clock. The time discrete sequence can then be written as

$$y_{\text{total},m} = \sum_{j=-M}^{N} b_j s(\varepsilon_{\text{tx},j} + \varepsilon_{\text{rx},j} - (m-j)T) \tag{5.4}$$

It can be decomposed into an ISI impaired signal and noise components. The noise components are due to TX jitter ε_{TX} and RX jitter ε_{RX} and are obtained by first order Taylor expansion of equation 5.4. This leads to the so-called EVN representation of jitter at the receiver sampler:

$$y_{\text{total},m} = y_{\text{isi},m} + y_{\text{txn},m} + y_{\text{rxn},m} \tag{5.5}$$

$$y_{\text{isi},m} = \sum_{j=-M}^{N} b_{m-j} s_j \tag{5.6}$$

$$y_{\text{txn},m} = \sum_{j=-M}^{N} b_{m-j} \left(h_{j-1} \varepsilon_{\text{tx},m-j+1} - h_j \varepsilon_{\text{tx},m-j} \right) \tag{5.7}$$

$$y_{\text{rxn},m} = \varepsilon_{\text{rx},m} \sum_{j=-M}^{N} b_{m-j} \left(h_j - h_{j-1} \right) \tag{5.8}$$

where $s_n = s(nT)$ and $h_n = h(nT + T/2)$. Note that here $y_{\text{txn},m}$ and $y_{\text{rxn},m}$ are the time discrete representation of $j_1(t)$, the jitter term of first order as also found in equation 5.1. The FIR filter and channel will take an effect on how transmitter jitter appears at the sampler while receiver jitter will only be scaled by a factor. Spectrally speaking, the transmit jitter experiences a coloring (filtering) process which is due to the channel frequency response and the FIR precoding efforts. Scaling of receiver jitter on the other hand is essentially given by the specific slew rate of the signal incident at the receiver. The interaction between slew rate and jitter can be illustrated more clearly by turning to a specific use case of the above formulas:

In the course of studying the effects of reference clock distribution the term *jitter amplification* was coined [7]. It refers to the fact that a clock source with a well defined jitter and distributed to a distant chip via a channel with frequency response $H(j\omega)$ will appear at the distant chip with different, potentially elevated, timing error. The problem increases with clock frequency, the number and severity of channel discontinuities and channel loss.

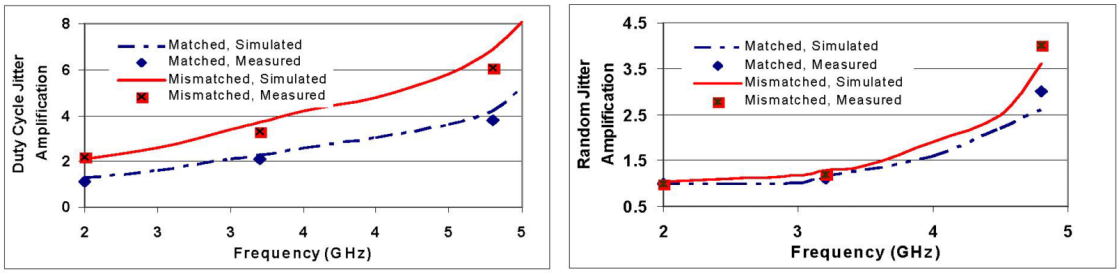

Figure 5.2: Duty cycle distortion amplification and random jitter amplification as measured and theoretically treated (a non-EVN formulation) in [7]

Figure 5.2 shows measurement of the original paper and displays the random jitter amplification factor in dependence of the channel loss at the clock frequency. With the formalism above, the situation can be modeled [33] by first setting $\boldsymbol{a}^T = \pm[-1,+1,-1,+1,...]$ which yields

$$y_{\mathrm{cn},m} = 2 \sum_{j=-M}^{N} (-1)^j \, h_j \, \varepsilon_{\mathrm{tx},m-j}$$

for the EVN of the clock jitter seen at the output of the distribution channel. Repeating the argument from above which relates single bit, step and impulse response, the zero-crossing slope of the clock signal at the channel output can be given by [33]

$$s_{\mathrm{clk}} = 2 \sum_{j=-M}^{N} (-1)^j h_j$$

from which the timing interval error at the channel output can be computed to

$$\varepsilon_{\mathrm{cn,out},k} = \frac{y_{\mathrm{cn},m}}{s_{\mathrm{clk}}} \tag{5.9}$$

With the so obtained sequence of timing interval error samples, the resulting RMS jitter can be calculated by forming the ACF of $\varepsilon_{\mathrm{cn,out}}$ and taking the value at $R_{\varepsilon\varepsilon}(0)$. As reported in [33] and displayed in figure 5.3, model and measurement are in quite reasonable agreement which is also of great importance for the alternative approach of link budgeting presented below. This example shows that by back-conversion of the EVN to a time interval error

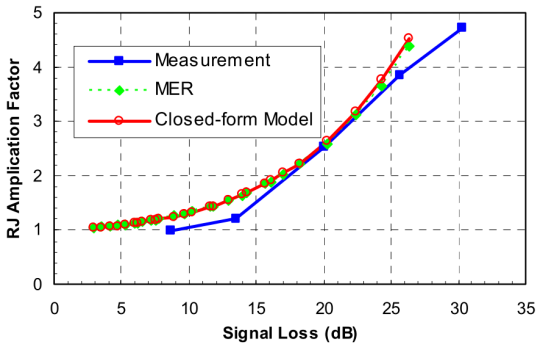

Figure 5.3: Random Jitter amplification versus channel loss at the clock frequency as measured (blue curve) and theoretically predicted by the EVN approximation (red curve) [33]

via the (clock) signals slew rate, a good estimate of the time interval error can actually be found. A much more intuitive view on this problem can be had in figure 2.2 in section 2.2.1. It shows the relation between timing and voltage error given by a signals slew rate near its crossing point. The Taylor series expansion holds as long as perturbations around a once chosen point in time (or voltage) remain fairly small and as long as the signal in question is not excessively nonlinear around the crossing point.

Returning to the link budgeting efforts, the transmitter and receiver jitter induced voltage noise seen at the sampler of the receiver is obtained just as in the example above - by computing the ACF of both $y_{\text{txn},m}$ and $y_{\text{txn},m}$. The autocorrelation functions can be compactly written ([60]) as

$$R_{y_{\text{txn/rxn}}y_{\text{txn/rxn}},m} = \bar{w}^T \overline{\boldsymbol{S}}_{\text{tx/rx},m} \bar{w}$$

with

$$\overline{\boldsymbol{S}}_{\text{tx},m} = E_a \sum_{j=-M}^{N} \sum_{k=-M}^{N} \overline{\boldsymbol{I}}_{m+j-k} \begin{bmatrix} h_{j-1} & h_j \end{bmatrix} \begin{bmatrix} R_{\varepsilon_{\text{tx}}\varepsilon_{\text{tx}},\, m+j-k} & R_{\varepsilon_{\text{tx}}\varepsilon_{\text{tx}},\, m+j-k-1} \\ R_{\varepsilon_{\text{tx}}\varepsilon_{\text{tx}},\, m+j-k+1} & R_{\varepsilon_{\text{tx}}\varepsilon_{\text{tx}},\, m+j-k} \end{bmatrix} \begin{bmatrix} h_{k-1} \\ h_k \end{bmatrix}$$

and

$$\overline{\boldsymbol{S}}_{\text{rx},m} = E_a \cdot R_{\varepsilon_{\text{rx}}\varepsilon_{\text{rx}},\, m} \sum_{j=-M}^{N} \sum_{k=-M}^{N} \overline{\boldsymbol{I}}_{m+j-k} \begin{bmatrix} h_{j-1} & h_j \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} h_{k-1} \\ h_k \end{bmatrix}$$

where E_a is the average transmit alphabet energy and thus dependent on the spectral content of the bit stream $\boldsymbol{a}$ sent and the output voltage swing V_{pp} of the transmitter. $R_{\varepsilon_{\text{rx/tx}}\varepsilon_{\text{rx/tx}}}$ are the autocorrelation functions of the receiver and transmitter PLL time interval error and $\overline{\boldsymbol{I}_k}$ is the identity matrix whose columns are barrel shifted right by k positions. The magnitude of $R_{\text{yy, TX/RX}}(0)$ is the measure of non-correlation in the resulting signal (due to the Whiener-Chintchin theorem) and is thus the EVN RMS noise magnitude due to

jitter at and in the vicinity of the crossing points of the signal. ISI is computed at the sampling point of the receiver in the middle of the eye when only considering equation 5.6 for a fixed m. However, as link budgeting algorithms seek to find both the eye height and width for a particular bit error level, it is common practice to calculate $y_{\mathrm{isi},m}$ at numerous different points for the worst case bit pattern of the alphabet $\boldsymbol{a}$ (which needs to be found by appropriate algorithms). The resulting vector $\boldsymbol{y}_{\mathrm{isi}}$ can then be interpreted as the two dimensional trajectory in the voltage versus time plane, whose probability of occuring (the z value at the location (t,v) or color code in the resulting eye diagram, see figure 5.7) is that of the pattern itself. Convolution with the PDFs of y_{txn} and y_{rxn} then gives the appropriate eye width and height. This convolution of an effective voltage noise with a time dependent voltage function will then produce the equivalent timing jitter at the signal transition (via the signals slope) and will else lead to a reduction of the eye height as is the case for ordinary voltage noise. More recent approaches such as in [44] delegate the construction of the *statistical eye* back to the time domain simulator which then also captures other deterministic influences on eye height such as duty cycle distortion or static, uncorrected offsets. Only the EVN process is then left for the numeric post processing domain thus consequently separating deterministic from random effects for increased simulation performance. In this case, simulation time needs to be long enough to either capture the entire alphabet or a preanalysis needs to be used to ensure inclusion of the worst case bit sequence.

The filtering effect of the CDR is modeled in [60] and [44] by a Markov chain statistical description of the control loop. The various positions of the sampling clock with respect to the input data are conceived as *phase states* and the probability to transition between every two phases can in general be computed from the probability density function of the receiver input signal jitter. The resulting output will be a steady state probability function of sampling at a particular location of the eye diagram. This probability can then be convoluted with the statistical eye to incorporate the effects of CDR filtering on the resulting eye diagram.

As the previous equations indicate, the matrices involved in computing the noise sources will grow big rather fast. However, after having seen all possible combinations of N+M bits, looking at more alphabet codes (which may have substantially longer sequences) will not change the result with respect to the significant bit error rate metric. On the other hand, all we are interested in are the proper combinations producing the few worst case responses. Also, it is the incoporation of the FIR equalization technique which makes the equations look very complex in addition to the apparent convolution process that every single equation describes. It seems tempting to move the description back to the frequency domain wherever possible - a feat that seems possible as indicated by the common mathematical basis as given by equation 5.1 and as will also be demonstrated by the clock jitter amplification example in section 5.3. A further motivation for this path is

the integration of other equalization concepts such as the CTLE into the analysis. They can be made part of the channel response which can then be dubbed *system response*. An impulse and single bit response can be found for this composite channel by performing a transient simulation of the entire system omitting all sources of noise. The impact of these voltage and timing noise sources on the final eye as well as the additional noise due to active components in the system channel can and should then be included separately.

While it may seem tempting at first to simulate the system with a complete sequence of the intended transmission alphabet, even with an entirely RNM modeled serializer system this may consume a substantial amount of time. Additionally, this procedure is ineffective as it evidently carries out the same mathematical task as described above, only this time in a "time unrolled" fashion. Luckily, equation 5.6 hints at its strong resemblance with a statistical approach that allows us to reduce the time based simulation efforts to one (or a few) single bit responses: the peak distortion algorithm (PDA) as presented in the next section. On the one hand, this allows us to handle all deterministic effects in time domain where they can be simulated efficiently. On the other hand, we can treat all timing (phase) noise sources in the frequency domain which includes the jitter amplification process as well as the impact of the CDR circuit on the final eye diagram using its linearized, phase space model. Also, dimensioning and power consumption of the receiver analog frontend is strongly depending on the size of the receiver samplers and their properties. Current approaches use best known estimates for intrinsic noise, sensitivity and aperture (bandwidth) of the samplers. Thus, overconstraining is unavoidable and a thorough budgeting effort which weighs the impact of comparator design choices on overall performance and power can not be realized. The procedure presented in section 5.3 will therefore take the sampler properties fully into account.

A final remark on a general frequency based solution according to equation 5.1: While it is of course possible to obtain both $S_{\varepsilon\varepsilon}$ - the power spectral density of the timing error function, and S_{xx}, the power spectral density of a given transmission alphabet with a decent computation effort, the actual spectral densities S_{x} and S_{ε} require full knowledge of the time signal in order to be unambiguous. If we had this information, the jitter term to first order could easily be computed to $j_1(j\omega) = \varepsilon(j\omega) * j\omega B(j\omega)$. With only the power spectral densities known, the Schwarz inequality tells us that (convolution is based on element-wise multiplication) $|j_1(j\omega)|^2 \leq |\varepsilon(j\omega)|^2 * j\omega|B(j\omega)|^2$ which thus only gives an upper bound.

5.2 The Peak distortion analysis algorithm

Over the past years, a multitude of different techniques has been developed to allow a quick assessment of voltage and timing margins in multi-gigabit serializer systems including but not limited to the ansatz above. The central challenge is the same for both pure analog and real number model based simulations: In the absence of random perturbations, the system

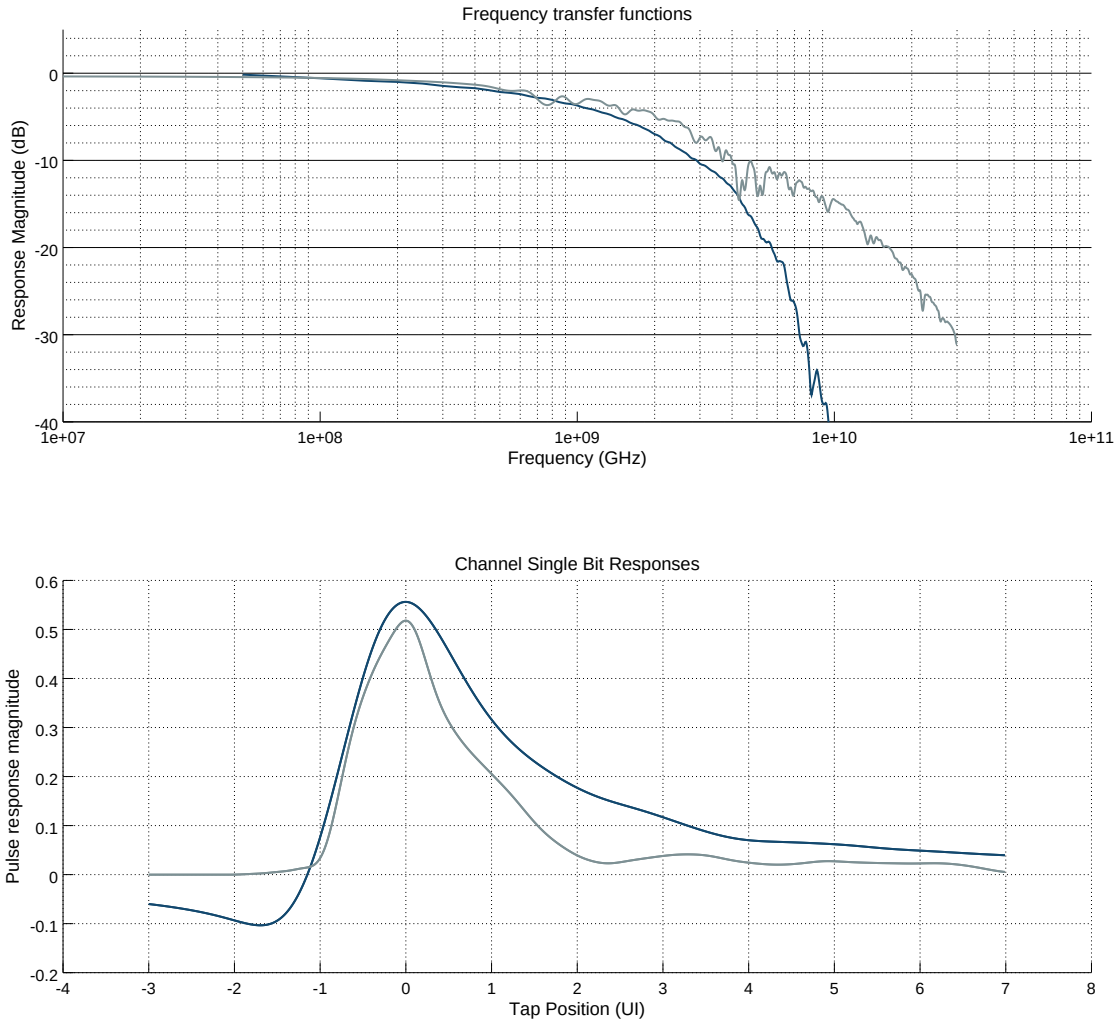

Figure 5.4: Exemplary channel frequency responses and their single bit responses to a stimulus with a unit interval of 100 ps.

of figure 2.1 is only comprised of LTI building blocks (except for the DFE that is). Its response to a given input sequence of bits can therefore be found by transient simulation (ergo: convolution of the system impulse response). To determine the worst-case margins of such a system, however, it is not sufficient to simulate a finite sequence of N random bits or to interpret the margins of a lone single bit pulse. To see this, the SBRs for two particular channels are plotted in figure 5.4 along with their frequency responses.

Due to the linear and time invariant nature of the channels, we may think of a sequence of transmitted bits leaving a channel as being equal to a superposition of multiple SBRs. This is actually the situation as described by equation 5.6 albeit not limited to the sampling

point at the receiver itself. As can be seen in figure 5.5, the SBR evoked by a bit sent at time instant m will overlap with the one originating from the bit sent at $m + 1$. Channels with strong high frequency losses or with multiple reflections within the channel will show single bit responses of long duration. Finding the voltage and timing margin requires to find the very bit sequence which minimizes timing and voltage margin for the single bit response in question. It is very unlikely that a random sequence of N bits will contain the worst case bit pattern and it is thus clear from figure 5.4 why a lone pulse will never produce the worst case result for channels with strong losses or reflections.

As described in the previous chapter, transient simulations by themselves would require too much simulation time due to the vastly different time constants and therefore bandwidths of the components in the system. This problem is not even alleviated much by resorting to more discrete simulation models such as those described in chapter 4. The alternative budgeting approach presented here relies on both transient simulation as well as statistical analysis provided by the PDA algorithm. PDA was first proposed by Proakis [52] and further developed and analyzed thereafter [6]. Usually, its starting point is the SBR of a channel which can be obtained by transient simulation. The central idea of the algorithm presented in this text however is to capture a multitude of deterministic effects as well, such as asymmetry in rise and fall times of the transmitter (which was also done in [44]), duty cycle distortion, reflections, residual offsets but most notably: the entire system channel itself consisting of FIR filter, CTLE and even the DFE. Therefore the SBR is extracted from a transient simulation which may either be an analog (SPICE) or real number model based (RNM) simulation of transceiver *and* channel.

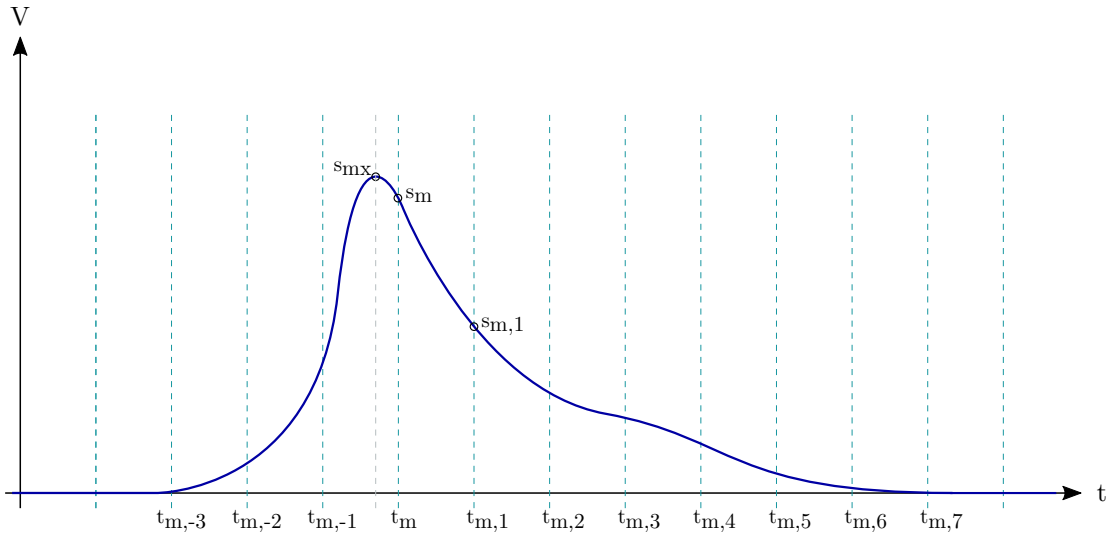

Figure 5.5: A single bit response with $t_m \neq t_{mx}$ and $M = -3$ and $N = 7$.

The SBR $s(t) = h(t) * p(t)$ from transient simulation is discretized by sampling

$$s(t) \rightarrow \boldsymbol{s} \qquad \text{with} \qquad s_m = s(t_m) \qquad \text{such that equivalently} \qquad s(t) = \sum_m s_m \delta(t - t_m) \qquad \text{holds}$$

with a resolution of $dt = t_{m+1} - t_m$. While in RNM simulations, dt is explicitly defined and chosen according to considerations presented in chapter 2 it may vary for analog simulations. A VerilogAMS sampling module on the other hand constrains the time step for the signal it probes in the analog simulator. For the same reasons as with RNM simulations, the time step in the transient simulation may be chosen larger than required in this algorithm. In both cases resolution can then be increased by cubic interpolation in the openMGT modeling framework (OCM) domain. This may be needed for further post processing (see section 5.3.6). The SBR is then truncated to the region of interest. The parameter $\boldsymbol{vr}_{\text{pda,thr}}{}^{\ddagger}$ is used to define how large the contribution of a sample of the discrete SBR s_m has to be compared to its maximum s_{mx} at t_{mx} so that it is considered relevant. The search is performed beginning at both start and end index of $\boldsymbol{s}$ toward the center in order to capture all potential reflections in the response. With a datarate R leading to a bittime (unit) interval T, the number of pre- and postcursor items will then be given respectively by

$$M = \text{floor}\left(\frac{t_{\text{mx}} - t_{\text{s}}}{T}\right) \qquad \text{and} \qquad N = \text{floor}\left(\frac{t_{\text{mx}} - t_{\text{s}}}{T}\right)$$

where t_s and t_e denote the first and last tap of significance in the SBR.

For convenience in the following, let us first define

$$s_{m,n} = s_{m+\bar{n}} \qquad \text{where} \qquad \bar{n} = n \cdot \frac{T}{dt}$$

As can be seen from figure 5.5, this describes the values of the SBR at bittime interval spacing $t_m + n \cdot T$. We call $n = 0$ the main cursor and all other values the pre- and postcursors to the time instant t_m. Note, that nothing explicit is being said here about the particular index m at which a potential receiver sampler actually would sample the signal itself. We will deal with this issue in section 5.3.6.

Now, the transition to a statistical description is being made: We would like to find the two dimensional signal probability distribution function $P(v,t)$ in dependence on time instant t and voltage magnitude v as it appears at the receiver sampler input due to the system channel. This density function can be conceived as the so-called eye diagram that would normally be recorded by an ideal, noiseless oscilloscope if it were sitting at the position of the receiver sampler. We therefore interpret the value $s_{m,n}$ as the voltage perturbation seen at the receiver sampler in addition to the main cursor voltage s_m at time instant t_m. The perturbation is caused by the symbol a_n sent n bittimes before or after the current bit of interest. With NRZ coding a_n can only take on the values ± 1. Thus, the

voltage perturbations $s_{m,n}\big|_{n\neq 0}$ will either increase or decrease s_m depending on the actual bit pattern sent prior to the current bit (postcursors) or thereafter (precursors). More formally, we write

$$s_{m,n} \xrightarrow{stat} p_{m,n}(v) = \begin{cases} \alpha, & v = s_{m,n} \text{ and } a_n = +1 \\ 1-\alpha, & v = -s_{m,n} \text{ and } a_n = -1 \\ 0, & \text{else} \end{cases}$$

which associates $s_{m,n}$ with the probability density function $p_{m,n}(v)$. Since the SBR is continuous in v, so is $p_{m,n}(v)$. It needs to be discretized for numerical processing. Therefore, a parameter $\boldsymbol{vr}_{\mathbf{pda_v res}}$[‡] is introduced which defines the binning interval for the voltages dv.

The parameter α is directly related to the transmission alphabet coding. For a PRBS the probability of transmitting and receiving a logic zero or one is equal. Therefore $\alpha = 0.5$. In order to get the total PDF P_m at a given sampling instant t_m, all possible combinations of bits at the various cursor instants have to be combined. With the convolution of two PDFs as given in 2.1 we define a cumulative convolution operator for all PDFs p_{mn} at a time instant t_m:

$$P_m = \sum_{n=-M}^{N} * \; p_{m,n} \equiv \left(\sum_{n=0}^{N+M-1} * \; p_{m,n} \right) * p_{m,M+N} \tag{5.10}$$

This recursive definition lends itself very well for a parallel implementation as the linear convolution operation is associative and can thus be executed in any order and with multiple subdivisions of the available $p_{m,n}$. The result of this operation will be the probability density function P_m of a signal passing through a particular voltage interval $v + dv$ at time instant t_m.

Figure 5.6 shows two different P_m for the light grey SBR shown in figure 5.4. m is chosen such that in one case, it is in between the SBRs maximum and the first precursor to the maximum. In the second case, m aligns with the maximum. At the maximum, the two distribution regions around $v = \pm s_{mx}$ do not overlap. In the absence of further noise sources, the statistical eye is considered "open". A potential sampler of sufficient resolution and sensitivity might readily convert the incoming analog signal into a valid stream of digital bits. In the first case, however, the chosen sampling point would be in very close proximity to the signal transition region. The two partial distributions overlap and no "eye opening" is observed at this instant in time.

If the various P_m are concatenated in order of their associated time instant t_m, a complete statistical eye can be produced (see figure 5.7). Thus

$$P_{\text{eye}}(v,t) = \sum_m P_m(v)\delta(t - t_m)$$

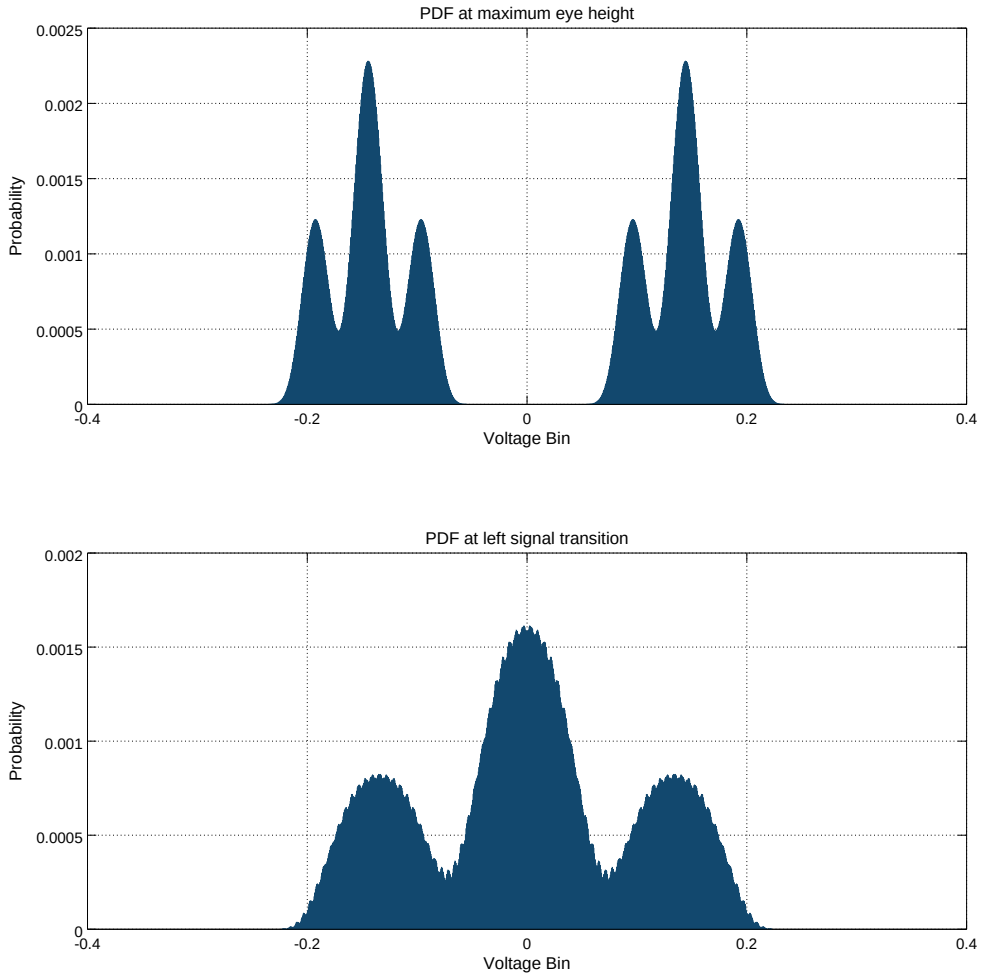

Figure 5.6: PDFs at the two sampling points described in the text

where in this case, of course, both v and t are of discrete nature. For a complete picture of the eye diagram, m is chosen such that it spans the region of $3 \cdot T$ around the maximum. Essentially, the algorithm emulates transient simulation and eye diagram overlay of the resulting waveform in one step. The number of emulated bits depends on the response length of the SBR. For an SBR with M significant precursors and N postcursors, the bit pattern length for a contribution to P_m is $K = M + N + 1$. Since all possible combination of bits and therefore all bit pattern contributions to P_m are analyzed, there are a total $S = 2^K$ patterns included in the analysis at any given point in time t_m. The number of bits a transient simulation would need to simulate to arrive at the same level of accuracy is

therefore given by

$$R = K \cdot S = (M + N + 1) \cdot 2^{M+N+1}$$

The three parameters of time resolution dt, SBR significance threshold $vr_{\mathrm{pda,thr}}$ and voltage binning resolution $vr_{\mathrm{pda_vres}}$ determine both the precision of the result as well as the total runtime of the algorithm. They must be chosen carefully so that runtime is not increased excessively while only a marginal benefit in accuracy is gained. The algorithm was implemented using the OCM framework in conjunction with the Octave software package (see chapter 4 for details on the interaction between simulations and post processing within the OCM framework).

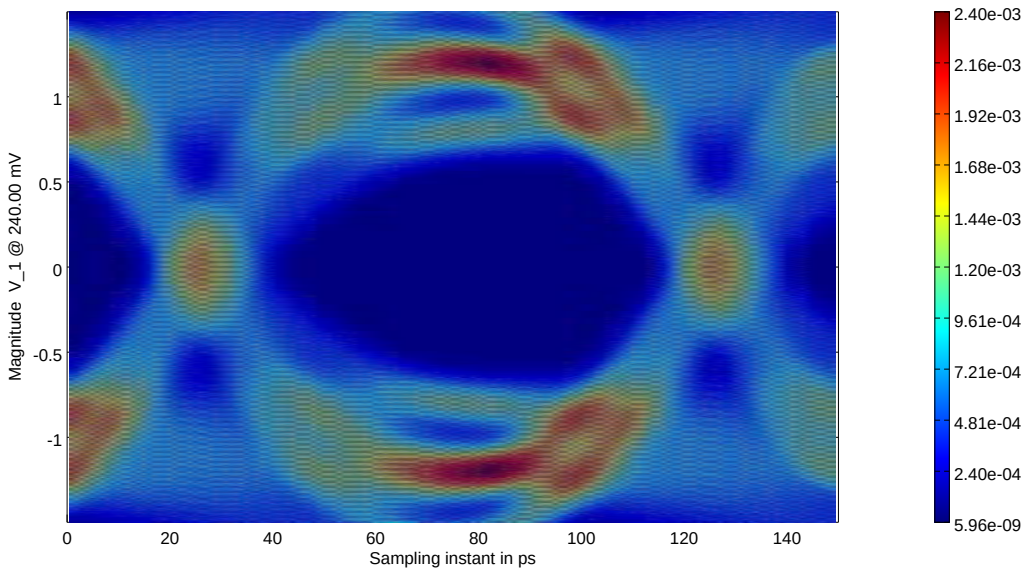

Figure 5.7: Statistical eye diagram resulting from peak distortion algorithm presented here. This figure

The example presented in figure 5.7 shows the deterministic eye of the SBR corrected by CTLE equalization for the light gray channel. Note that the columns are normalized to unity so that the probabilities as indicated by the colorbar to the right of the picture always refer to a time instant. The rows will therefore not be normalized and will require renormalization when used as deterministic PDFs. The analysis was carried out with a time resolution of 1 ps and a voltage resolution of about 600 uV. This lead to five pre- and 21 post cursors which therefore accounted for approximately $R = 3.6 \cdot 10^9$ statistically treated bits. Since the unit interval T is 100 ps in this case, the total transient simulation time to arrive at the same result would have been **360** ms accordingly. The algorithm requires a total computation time of merely **3.5** s in this case for a single SBR to PDA conversion

on a 12 core Intel Xeon E5-2630 system running at 2.3 GHz and 96 GByte of RAM (the memory footprint is marginal with a few tens of megabytes).

A disadvantage of the procedure described here is that the actual alphabet coding is not considered. It may actually improve the eye if worst case patterns are not contained in it. Links are often tested with PRBS patterns though and if they are held to be the measure of a system passing or failing a specifc BER target, searching for worst case patterns is the default approach. If the alphabet coding was 8B/10B for instance, the limited set of allowed sequences will lead to a modified value for α. There is however a way in which a future algorithm could be improved in this aspect: while still using equal probabilites for ones and zeros, the convolution process could be extended to filter out the disallowed sequences by tracking how a partial PDF p(v,t) is generated. If a specific sequence is not contained in the given alphabet, it could be omitted from further processing. Since there is a lot of lookup effort involved, this may, however, proof computationally intense.

The computationally efficient PDA algorithm presented here forms the foundation of the serializer system budgeting and analysis algorithm presented in the following.

5.3 The OCM link budgeting algorithm

This section introduces the budgeting algorithm as it is employed in the openMGT/OCM framework. It combines the benefits of peak distortion algorithm - statistical treatment and inherent parallelization - with the computational efficiency of treating random phase noise sources in frequency domain to first order (compare equation 5.1). Its core idea directly stems from the concept pursued with the entire openMGT framework: Separation of all deterministic system properties from all random perturbations and treatment of as many deterministic effects as possible within the transient pre-simulation while delegating more complex processes with larger time constants to the OCM/OCD post processing domain. While the statistical PDA algorithm characterizes deterministic aspects of a given transmission channel, there are the above-mentioned random contributions such as voltage noise and timing jitter which originate from random physical processes such as 1/f and thermal noise of the electronic system. System performance constraints originating from random contributions can quite generally be observed by a transient simulation. However, the likeliness of a specific event to occur in simulation is directly related to the probability function of the underlying process. In a multigigabit link analysis with target bit error rates of below 1e-12 and simulator time steps in the picosecond range, it is quite evident that even the most potent computing platforms will not be able to solve the problem with transient approaches within a sensible amount of time. Therefore the process of analyzing the link system is segregated into two distinct portions: Deterministic effects can and may be observed and checked by simulation approaches. The task of taking random

effects into account will entirely be placed inside the numeric modeling domain. This is true for both random voltage and timing uncertainties. Since the translatation between voltage and timing noise is always possible within the limits described earlier, the form used for budgeting depends on the availability of data from other simulations or on how specification metrics are defined (which usually depends on the available testequipment required for these meausrements).

Equalization settings provide a very good insight into how well metric constraints are chosen for a particular serializer implementation. Consider a worst case channel as defined by most modern protocol specifications: If it is used for budgeting and the equalization settings required are fairly moderate compared to the allocated equalization capabilites, the odds are very high that some part of the system was overconstrained. Often times, this may be the linear preamplifier in the receiver which incurs a significant power penalty for the whole transceiver. This an important reason why equalization training is part of the transient pre-simulation for the budgeting procedure. The other is of course the most decisive question whether or not the given system can ensure a specific target BER on the transmission channel in question.

With the physical transmission channel given, the PDA produces a probability density function of a specific voltage to be encountered at the input of a sampler at a particual point in time (the sampling instant). This two- dimensional PDF of the entire system channel lends itself well to be further processed by combining it with probability distributions of voltage and timing noise sources. The dominant noise source in broadband circuits is thermal noise (Johnson-Nyquist noise) of the constituent components. Techniques to evaluate the voltage noise added by a given electronic topology to its output signal with respect to its input are readily available and either analytical or simulation based. From the derived voltage noise power spectral density $S_{VV}(f)$ the total RMS voltage noise appearing at the output of the circuit can then be computed to

$$\sigma_n = \int_0^\infty S_{vv}(f)df$$

Due to strict separation of deterministic and random contributions (see figures 5.8 and 5.9), at least the deterministic, two dimesional eye PDF and the random PDFs can be assumed to have originated from entirely uncorrelated sources. This is not be the case for the PDFs of voltage and phase noise since all the components in either PDF that originate from the same buffer or the same power distribution would indeed be a source of correlation. In this sense, the final outcome of the budgeting procedure is actually conservative.

The deterministic effects which are covered by the openMGT real number models for budgeting are

- TX duty cycle distortion (DCD) due to static mismatches in the TX clock distri-

bution and potential PMOS/NMOS mismatch in the output driver which leads to asymmetries in rising and falling transitions $r(t)$ and $f(t)$

- The channel response $H(j\omega)$ as given by either an impulse response or S-Parameter file
- Transmitter and receiver termination mismatch Γ_s and Γ_l, which are mainly due to large polyresistor variations in advanced technology nodes and can be compensated by digitally adjustable resistors with a finite resolution and thus a finite error
- Static mismatches in the receiver analog frontend which even in the presence of calibration logic lead to a residual offset error Δ_{resoff}
- Sampler nonidealities as described in section 4.5 which lead to ISF $\Gamma(t)$, finite dc sensitivity χ_{dc} and intrinsic offset Δ_{off}
- Power distribution noise backaction due to the spectrally resolved current consumption of the system in cases where Z_{PDN} is known.

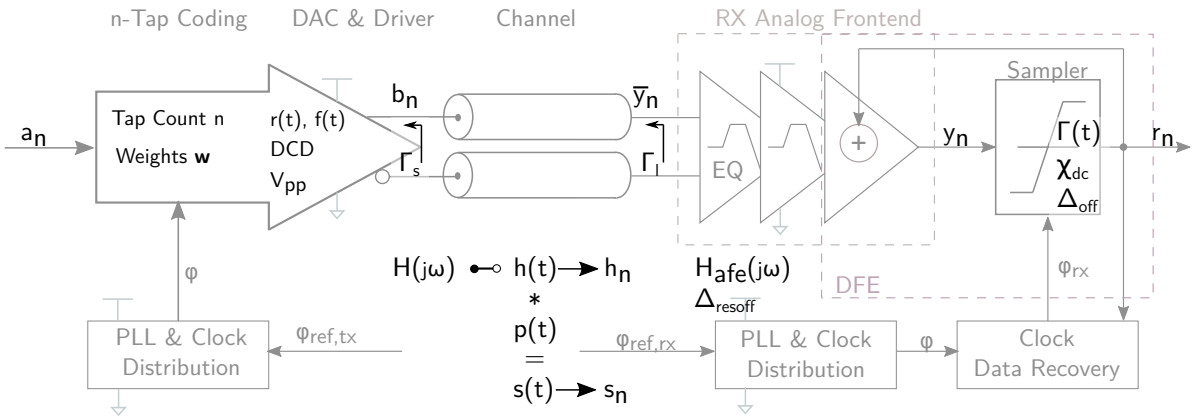

Figure 5.8: Deterministic effects and their respective subcomponent as used during OCM link budgeting

A transient pre-simulation will consist of

- Initializing offset correction vectors where applicable such that the residual offset error will be minimized. This is a shortcut to the calibration routines which are normally used to find a particular correction vector for a given stage. These loops are verified separately so it can be assumed here that the optimal code is actually retrieved during serializer startup.

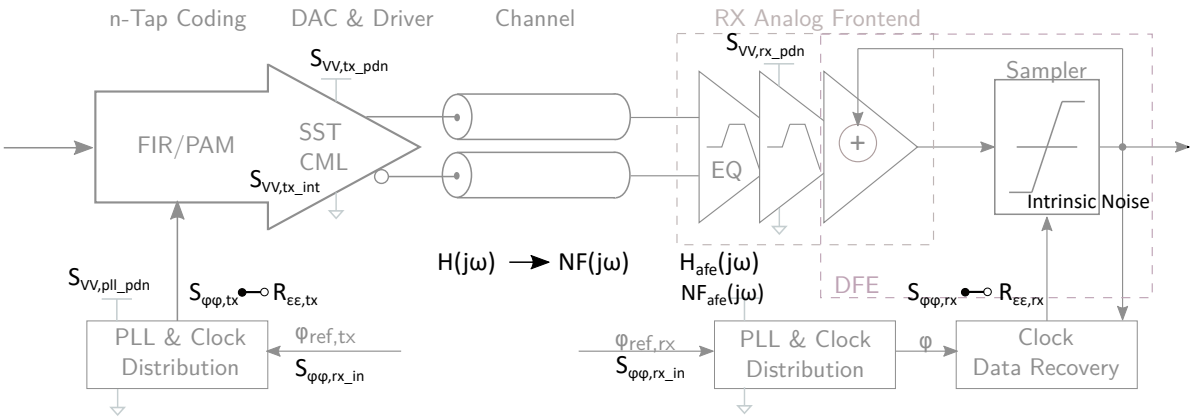

Figure 5.9: Random effects and their respective subcomponent as used during OCM link budgeting

- Running an equalization adaption algorithm which simultaneously optimizes the FIR, CTLE and DFE settings (see chapter 3). During transient simulation, it is also checked that the CDR circuit locks for the equalization settings found. In this way, the residual equalization error due to finite tap strength resolution (or pole resolution in case of the CTLE) is also taken into account.

- For every output bit phase, two pulse responses are generated. One transitioning from the logic high to the logic low state, the other vice versa. For a typical two phase output driver (i.e. DDR), this would result in four separate bit pulse sequences, for the quarter rate design analyzed in this work (see also chapter 6), a total of eight pulse responses need to be generated. Each pulse response needs enough lead-in time, such that no residual ISI perturbations of previous bits are present on the signal to correctly capture the lone SBR.

The pulse responses and the system constraints to all models including PLL, CDR and power delivery network are then exported to the OCM domain via the appropriate SystemVerilog DPI calls. Due to the implementation consistent modelling approach of the openMGT framework, the final performance evaluation can also be done with a schematic based system. The very same budgeting procedures described below can then be used to verify compliance. Since the openMGT framework uses the RNM model parameters as a basis for implementation validation, this information is also transferred to the OCM budgeting domain whenever necessary to derive the apropriate performance metrics. This is either done by ammending the RNM models directly with the respective OCM/DPI system calls or by keeping central specification information in a *global definitons file* which belongs to a particular verification test. Such global parameters include the actual bit unit interval $\boldsymbol{UI}^{\ddagger}$, the system temperature $\boldsymbol{T}_{\mathbf{env}}{}^{\ddagger}$, the resolution of phase noise

functions $\boldsymbol{df_{\phi}}^{\ddagger}$, the (random) digital test sequence used during analysis as well as the supply voltage level and (in verification runs) the technology corner used. As described in sections 4.4 and 4.5 the actual component modelling for channel and sampler is implemented in the OCM domain itself. In this case, the information required for postprocessing is already in place and consistent with the data used throughout the transient pre simulation.

Post processing is separated into seven distinct steps which may themselves be further broken down into substeps.

5.3.1 Power distribution

The budgeting procedure checks for the presence of a PDN impedance profile for PLL, RX and TX. This can either be an impedance mask or a more realistic impedance curve as generated by EM extraction of board and package together with the known characteristics of VRM and capacitances. In order to use the impedance profiles, a file describing the subcomponents spectral current consumption must also be supplied. This current profile can potentially be obtained together with the transient budgeting simulation itself, provided that all subcomponents of the system feature activity dependent power models or a full schematic based transient simulation has been carried out. In the absence of this information (which is typical for a predesign phase), the procedure reverts to the simple box or lowpass noise model specified by the global defintions file. How the information is being processed from this point onward depends on the actual subcomponent and is described in the respective sections below.

5.3.2 PLL phase noise spectral density for transmitter and receiver

Referring to section 2.2.4, it is possible to use and constrain parameters for the subcomponents of the PLL as a basis for further link budgeting. The budgeting procedure compiles the information from the global definition file and uses the equations described to produce the phase noise spectral density $S_{\phi\phi}$. They are used both for the transmitter and the receiver unless the phase noise properties of their respective clock distribution is known as well. In cases where system components are already designed, the more accurate, simulation based phase noise PSDs may and should be used instead of the model. The model, however can always be used as a compliance check for the actual implementation. Once the PLL phase noise spectral density $S_{\phi\phi,PLL}$ is computed, a global parameter $\boldsymbol{mag_{\phi,trsh}}^{\ddagger}$ is used to determine the *phase noise bandwidth* $\boldsymbol{\omega_{\phi,\mathrm{BW}}}^{\ddagger}$, i.e. the frequency range to be considered during phase noise analysis in all subsequent steps. Since there is also a strict separation between voltage and phase noise sources in the budgeting flow and we expect the major random contributions to phase noise in the system to mainly arise from the PLL itself, this limitation keeps the phase noise vector at manageable sizes.

5.3.3 Peak distortion analysis

The PDA algorithm is the central element of the budgeting procedure. As mentioned in section 5.2, the statistical algorithm is used to decrease the transient simulation time and to acquire an accurate estimate of the two-dimensional eye PDF at the sampler input and is as such a complete representation of $y_{\mathrm{isi},m}$ in equation 5.6 for all potential sampling points m. Here, we extend the idea of the intersymbol interference term. First, it also covers the residual signal deterioration which is due to imperfect equalization. Also, it covers the effects of residual offsets in all equalization stages. And finally, the deterministic effect of duty cycle distortion is also taken into account as part of the transient simulation. The resulting estimation thus not only absorbs the effects of the FIR weighting matrix $\overline{\boldsymbol{W}}$, but also captures the signal improvement efforts of CTLE and DFE without having to describe their effects in the time domain mathematically - literature on budgeting and system performance estimation usually does not cover these aspects at all. In addition, since the serializer system is simulated in conjunction with its calibration and equalization loops, the quality of their respective result is considered implicitly. This helps to choose the right dimensions for calibration vectors and filter constants for the respective control loops. Due to this great degree of coverage, a more appropriate symbol for the resulting quantities is $P_{\mathrm{eye,\,det},v,t_m}$.

One of the central TX and RX design parameters is the number of clock phases used for (de-) serialization. If the final retiming stage of the transmitter is double data rate and differential, it requires two clock phases. If a receiver is said to be a *quarter rate design*, it uses four distinct clock phases with its four data samplers. In order to capture the effects of duty cycle distortion, the resulting PDA needs to be calculated from equally many distinct single bit responses as there are clock phases in the transmitter. The distinct SBRs need to originate from the respective clock phases and will lead to differing, determinstic eye heights and widths. In order to capture asymmetries in transmitter rise and fall times or nonlinearities in the preamplification stages of the receiver, for every TX output phase both logic transitions will be simulated. For a quarter rate design for instance, this results in a total of eight single bit responses to be analyzed seperately. The resulting statistical eyes will be weighted equally and summed. Strictly speaking, the transmission alphabet would need to be factored in at this point as it may exhibit a statistical imbalance between the logic levels. Since the PDA itself, however is also calculated under the assumption of equal probabilities for all logic levels, this notion is perpetuated here. The requirement of calculating multiple statistical eye diagrams is also a major reason for the parallelization efforts during the development of the OCM PDA algorithm.

Figure 5.10 depicts the result of a statistical system channel analysis with the serializer system as presented in the next chapter. The system does not perform proper equlization. As a result, the statistical eye is almost closed, even in the absence of superimposed voltage

and timing noise. The linear color scale as used by figure 5.10 suggests a far greater eye opening than actually present. A logarithmic color scale for the same situation (figure 5.11) reveals the degree to which the eye is closed much more drastically.

Figures 5.12 and 5.13 show the same situation again with linear and logarithmic color scale, only this time with adjusted FIR and CTLE settings. The equalization may not be optimal but already produces a drastically better statistical eye.

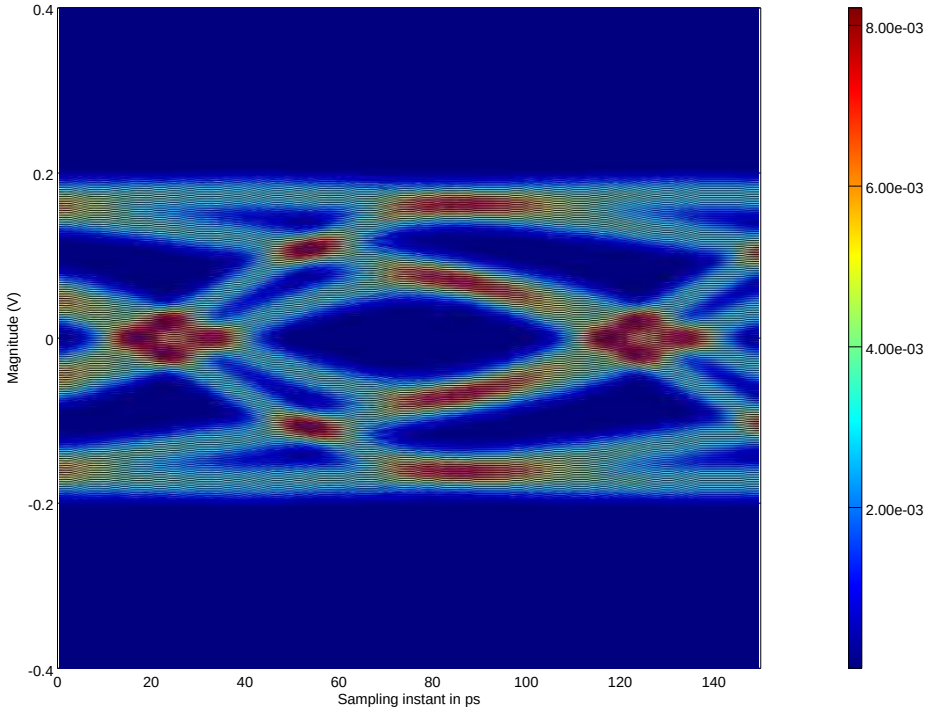

Figure 5.10: *Unequalized* statistical eye of the light gray channel in figure 5.4 resulting from real number model simulation with the serializer system presented in chapter 6 at 10 Gbps and with linear color scale

5.3.4 System channel and jitter amplification in CDR based systems

With $P_{\text{eye, det},v,t_m}$ replacing the intersymbol interference term in previous approaches to link budgeting, a new approach on how to compute $j_1(t)$ (more precisely equations 5.7 and 5.8) needs to be found. This is primarily due to the impact of the CTLE which cannot be described as elegantly as an FIR filter with a single matrix for instance. However, this component now is a part of the deterministic eye PDF in which we intend to integrate the systems noise properties. Also, as already mentioned before, the rather large matrices and convolution efforts indicated by equation 5.7 and 5.8 may result in substantial computation

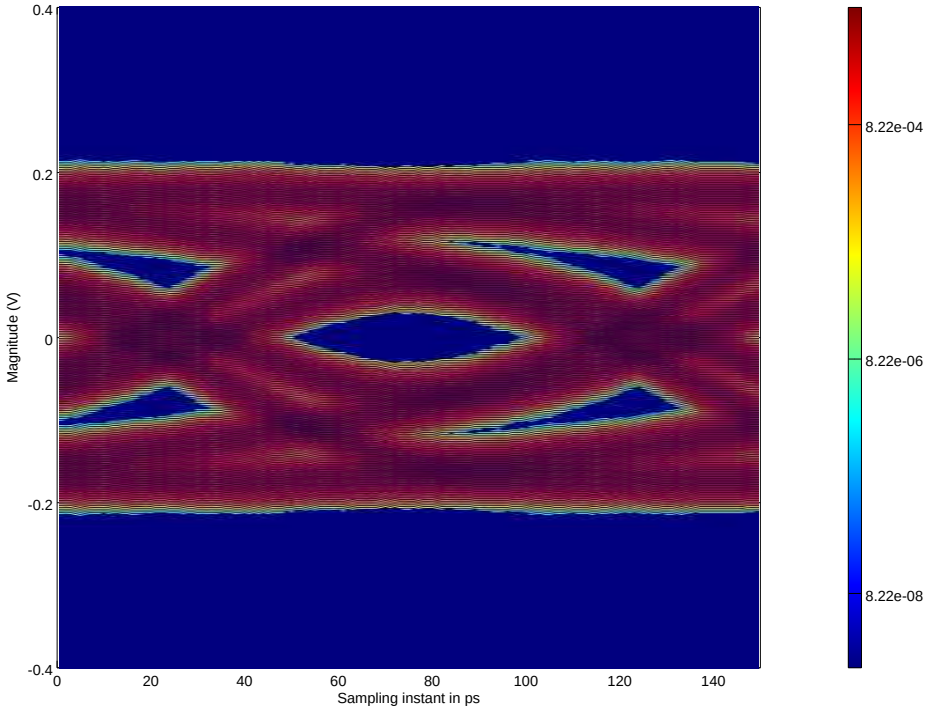

Figure 5.11: *Unequalized* statistical eye of the light gray channel in figure 5.4 resulting from real number model simulation with the serializer system presented in chapter 6 at 10 Gbps and with logarithmic color scale

efforts and memory requirements, especially for very long channel responses.

A description in frequency space could therefore proof to be a viable option. In the context of clock channel jitter amplification as discussed in section 5.1, this has already been shown to be totally equivalent by previous investigations[53]. Due to equation 5.1, this does not come as a surprise.

Thanks to the equalization settings reported by the transient simulation and the equivalence of the models used in simulation and postprocessing (or the validation of implementation against the models), the frequency transfer functions of FIR and CTLE can generally be derived. We can thus compute the overall system channel response and use this as a sensibility check for the equalization settings as applied by the various adaption control loops (see figure 5.14).

Care must be taken here, as the system transfer function for the phase noise seen at the CDR sampler input might *not* be shaped by the DFE transfer function. A DFE with effect on phase noise would also need to equalize the signal seen at the CDR phase detector input (or equivalently: the RX edge samplers). Since this may have delicate repercussions on CDR performance (after all, it is not entirely clear how an 'equlized' signal slope is supposed to look like and how it affects the sampling point of the CDR itself), the

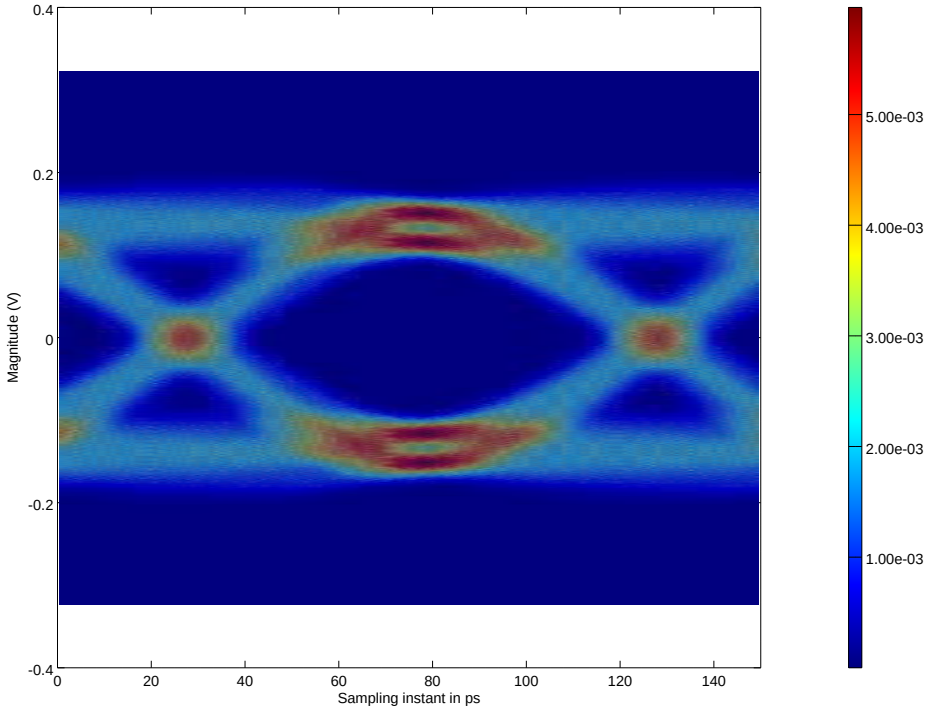

Figure 5.12: *Equalized* statistical eye of the light gray channel in figure 5.4 resulting from real number model simulation with the serializer system presented in chapter 6 at 10 Gbps with linear color scale

phase detector is usally supplied with the signal as output by the receivers analog frontend. This aspect of system design will be an interesting future topic for which the openMGT framework may be very useful and is not covered further in this text since the serializer design to be analyzed in chapter 6, does indeed not correct the edge path (see figure 6.3).

With a frequency description of the system channel available, the key ideas of this approach can now be developed:

first, instead of using the ACF of the PLL jitter input to the TX, we use its Fourier transform - the power spectral density. $R_{\varepsilon_{\mathrm{tx}}\varepsilon_{\mathrm{tx}}}(t) \circ\!\!-\!\!\bullet\, S_{\phi\phi,\mathrm{tx}}(\omega)$ Although potentially recomputable to a frequency power spectral density (referring it to the oscillation carrier), the ansatz in [53] suggests a more elegant alternative which directly uses the one sided phase noise spectral density. This view is particularly helpful here as the second idea directly benefits from this thought:

We view the clock recovery circuit as a special kind of phase locked loop. Its major property is its capability of working with multiple, divided versions of the same reference clock embedded in the data stream at once in order to recover the phase and frequency offset between the embedded clock and the local oscillator. This ensemble of frequencies results from the different patterns and run-lengths of logic ones and zeros in the incoming

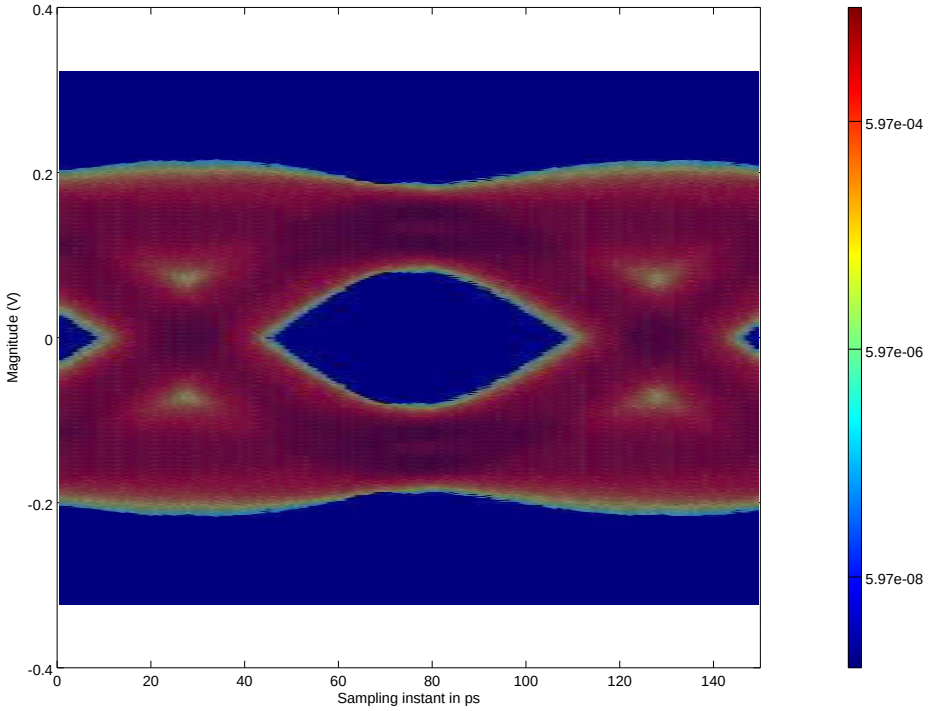

Figure 5.13: *Equalized* statistical eye of the light gray channel in figure 5.4 resulting from real number model simulation with the serializer system presented in chapter 6 at 10 Gbps with logarithmic color scale

data stream and thus ultimately depends on the power spectral density of the transmission alphabet. Again, we restrict the discussion and analysis to pseudo random sequences for which the PSDs can be derived analytically. When using more complex line or alphabet coding schemes, PSDs can also be computed by fast real number based simulations. The phase noise properties of all these TX generated carriers are initially the same - after all, the serialization takes place with the ever same TX reference clock. Only the way in which the channel and equalization act upon their phase noise spectrum will be vastly different.

Third, we interpret the power spectral density as a probability density function: a lower frequency embedded in the data stream has far more power density associated with it than a frequency close to the baud rate. On the other hand, this very same low carrier frequency leads to only a small count of transitions per unit time at the clock data recovery phase detector. The power spectral density PDF is therefore weighted with the carrier frequency and normalized to unity again. This gives the relative importance of the phase noise properties at and around a given carrier frequency.

According to [53], the sinusoidal jitter amplification factor of a perturbing sideband

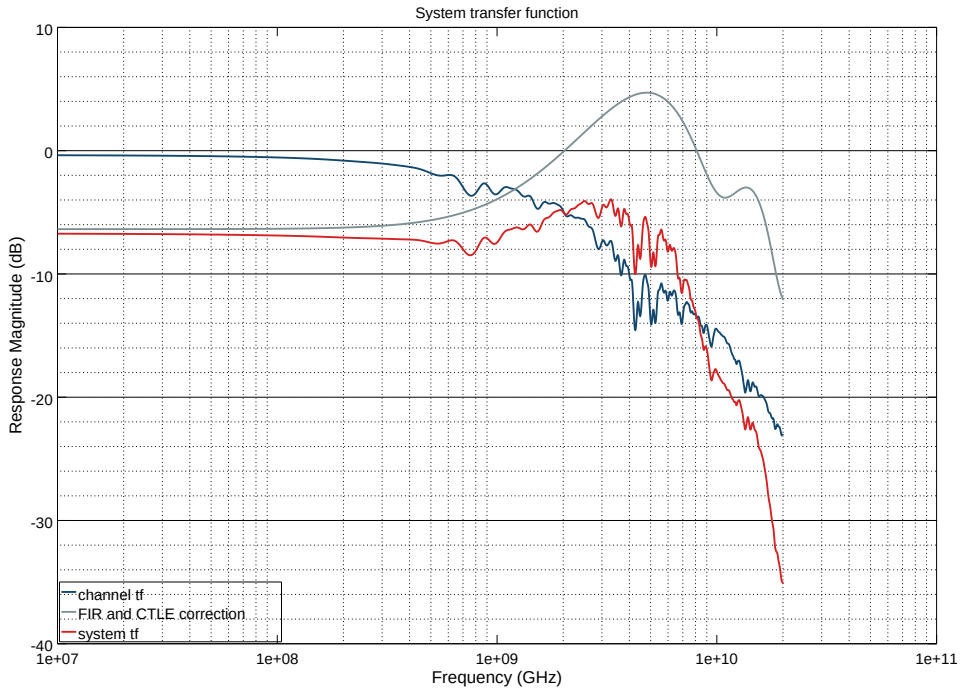

Figure 5.14: *Equalized* system channel response of the light gray channel in figure 5.4 resulting from real number model simulation and equalization adaption with the serializer system presented in chapter 6 and the statistical eye as presented in figure 5.13

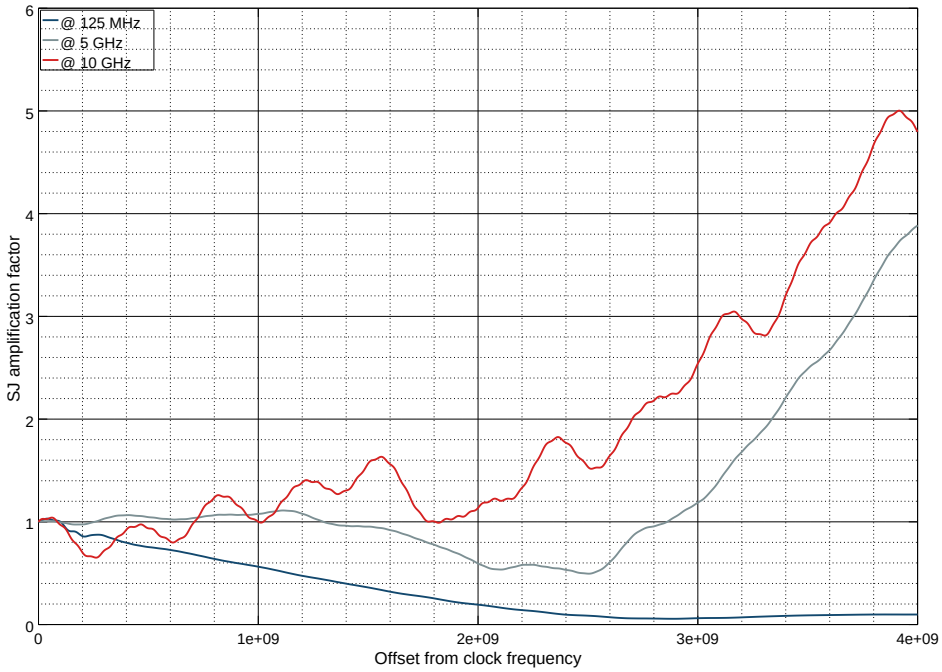

Figure 5.15: Sinusoidal jitter amplification factor for the dark grey channel in figure 5.4 and for various clock center frequencies

signal to a center frequency at ω_0 can be calculated to

$$F_{\mathrm{SJ}}(\omega, \omega_0) = \frac{1}{2}\left|\frac{H(\omega+\omega_0)}{H(\omega_0)} + \frac{H(\omega-\omega_0)}{H(-\omega_0)}\right| \tag{5.11}$$

with the special case of duty cycle distortion amplification

$$F_{\mathrm{DCD}} = \frac{1}{2}\left|\frac{H(2\omega_0)}{H(\omega_0)} + \frac{H(0)}{H(-\omega_0)}\right|$$

Note however, that DCD is considered deterministic and thus is covered by the transient simulation. For low carrier frequencies with broadband noise, the channel constitutes a low pass filter effectively removing high frequency phase noise content (compare the dark blue curve of figure 5.15). At higher frequencies however, well beyond the channel cutoff frequency, the amplification of jitter originates from the stronger attenuation of the carrier compared to the noise sidebands at lower frequencies. The effect is stronger for channels with high loss or strong reflections, i.e. for channels where the transfer function changes more drastically with frequency. Since it is not very intuitive to describe amplification without actually having some kind of gain definition, Rao also specifies a random jitter amplification factor F_{RJ}. It relates the phase noise seen at the channel input to the phase noise observed at the output. Since Rao uses all white noise for this purpose to arrive at a compact equation, this factor is of limited use here and also conveys a wrong picture once the spectral content of phase noise incident on the channel actually is known. For the random jitter amplification, which is just a means of visualization here, this text suggests a more general viewpoint:

$$F_{\mathrm{RJ}}(S_{\phi\phi}, \omega) = \frac{\int_0^\infty F_{\mathrm{SJ}}(\overline{\omega}, \omega) S_{\phi\phi}(\overline{\omega}) d\overline{\omega}}{\int_0^\infty S_{\phi\phi}(\overline{\omega}) d\overline{\omega}}$$

which of course includes the above mentioned special case of white noise when proper integration bounds are used. Due to the low pass colouring process usually exhibited by meaningful phase noise spectral densities or the usual frequency bounded all white Gaussian noise definitions, the integration range here can safely be extended to infinity. For the light grey channel in figure 5.4 and the PLL phase noise spectral density in image 2.9 the resulting function is plotted in figure 5.16.

From the figure, it is also evident why there is no risk of the integral being unbounded. The phase noises densities usually drop much faster than amplification factors rise over frequency for reasonable channels. As can be seen, the comparably low attenuation of the light gray channel only leads to significant jitter amplification factors above unity for a white power spectral density composition. The rather narrow band PSD of the low jitter PLL will only experience slight amplification at higher frequencies which are mainly

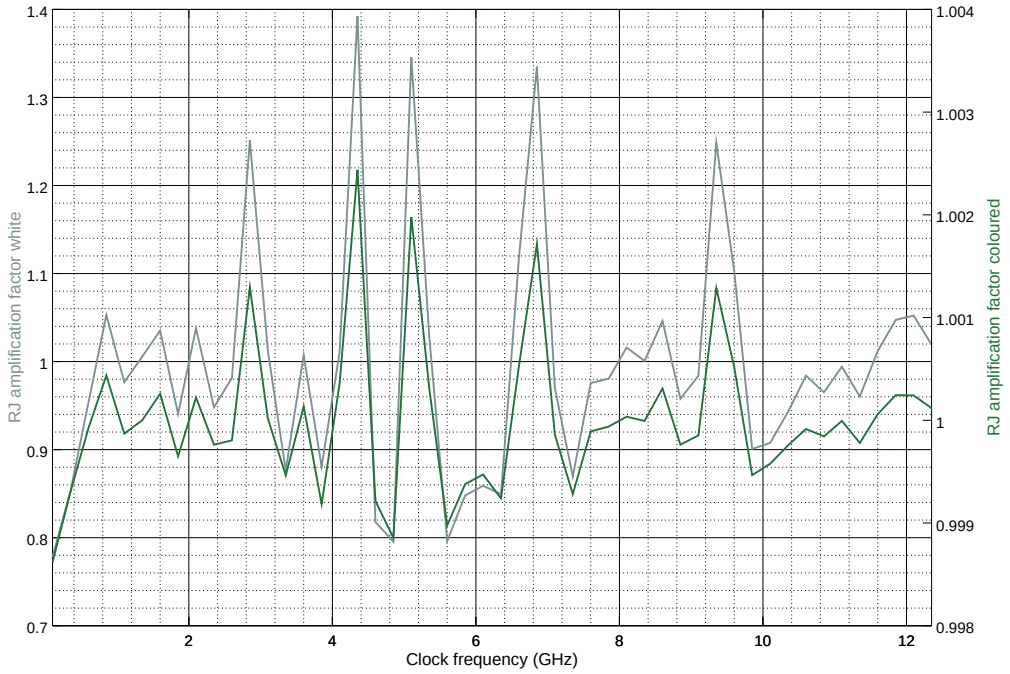

Figure 5.16: Random jitter amplification factor for the light grey channel in figure 5.4 and white noise as well as coloured noise of an example very low jitter PLL as given in figure 2.9

due to the discontinuities in the channel. The problem will, however, grow with stronger attenuation in the transmission channel and higher data rates.

To calculate the phase noise colouring process of the system channel for each carrier ω_0 we can then use

$$S_{\phi\phi,rx}(\omega, \omega_{0,m}) = F_{\mathrm{SJ}}(\omega, \omega_{0,m}) \cdot S_{\phi\phi,tx}(\omega)$$

The final *statistical* RX sampler input noise PSD can finally be calculated by weighing all carrier phase noise PSDs with their central magnitude as given by the alphabet coding PSD $S_{\mathrm{AA}}(\omega)$

$$S_{\phi\phi,\mathrm{rx\_stat}} = \sum_m S_{\mathrm{AA}}(\omega_{0,m}) \cdot S_{\phi\phi,rx}(\omega, \omega_{0,m}))$$

As derived in section 3.4 the PSD of a pseudo random NRZ sequence is a Dirac comb in frequency space with an envelope as given by equation 3.13 and a comb spacing as given by equation 3.14. The frequency resolution of the phase noise functions needs to be constrained to manageable lower bounds. Thus, calculation of the statistical phase noise is actually restricted to the base frequency vector and the resolution defined by the OCM budgeting system, irrespective of the alphabet coding used. If the frequency resolution

$\boldsymbol{df_{\text{phase}}}^{\ddagger}$ is not choosen too liberally, the channel properties change only very mildly in its vicinity. A further increase in resolution (or number of carriers) would therefore have little to no impact on the resulting phase noise spectral density.

The joint treatment of equalization and channel to compute the resulting phase noise properties at the receiver may seem to claim a complete 'recoverability' of the initial TX output phase noise spectrum by equlization. This would be somewhat reminiscent of asserting that equalization can restore an initial signal to noise ratio even though all power dissipating stages will inevitably add to the overall noise level. Actually, the reason for this inconsistency of sorts is due to the separation of voltage and phase noise. The excess noise of active amplification stages in the analog frontend is not attributed to the phase noise but rather mapped to it during the final CDR phase noise calculation (see below). This effectively increases the phase noise level again and resolves this situation.

5.3.5 System voltage noise estimation

The OCM budgeting framework supports to model the power distribution, noise sources and noise susceptibility functions either through simple low pass or box models as in the case of the PLL or it accepts data extracted from simulations of schematic based implementations. For very simple noise susceptibility modelling, first order high pass models with characterisitic onset frequencies are used. The voltage noise output at the transmitter then computes to

$$S_{\text{VV,tx}} = H_{\text{hp}}(f_{\text{psrr,tx}})S_{\text{VV,tx,pdn}} + S_{\text{VV,tx,int}}$$

where $\boldsymbol{f_{\text{psrr,tx}}}^{\ddagger}$ denotes the onset frequency of failing power supply rejection in the transmitter output buffer (in case the TX noise susceptibility is not supplied by a data file) and $S_{\text{VV,tx,int}}$ is the power spectral denstiy of the TX device noise sources.

The channel added noise on the other hand entirely depends on its loss properties and is computed under the asumption of a matched termination in trasmitter and receiver alike. It can be shown [51] that the channel added noise power spectral density can be computed according to

$$S_{\text{VV,c}} = 4Z_0 k_{\text{B}} T\, s$$

where s is the frequency dependent noise figure of the channel given by $s = -10log_{10}(|S_{21}|)-1$, i.e. via the forward transmission parameters of the channel.

The resulting noise at the receiver input will therefore be

$$S_{\text{VV,rx,in}} = S_{\text{VV,tx}}|S_{21}|^2 S_{\text{VV,tx}} + S_{\text{VV,c}}$$

The noise added by the receiver and thus the total spectrally resolved voltage noise as

appearing on the sampler input can then be caculated to

$$S_{\mathrm{VV,rx}} = H_{\mathrm{hp}}(f_{\mathrm{psrr,rx}})S_{\mathrm{VV,rx,pdn}} + S_{\mathrm{VV,rx,int}} + |H_{\mathrm{rx}}|^2 S_{\mathrm{VV,rx,in}}$$

Actually, $S_{\mathrm{VV,rx,int}}$ would depend on the chosen equalization setting H_{rx}. This could potentially be considered in the future. For the purpose of budgeting here, it shall represent the worst case added receiver noise.

In addition to the transmitter, channel and receiver induced noise contributions, one of the most important aspects is the sampler self induced noise (see section 4.5). The sampler noise simulation results are not spectrally resolved and therefore require the input referred sampler RMS voltage noise to be considered spectrally white finally giving

$$S_{\mathrm{VV,sampler,in}} = S_{\mathrm{VV,rx}} + S_{\mathrm{VV,sampler,self}}$$

Figure 5.17 gives an exemplary overview of the various noise sources and shaping processes with the final voltage noise PSD at the sampler input on the lower right.

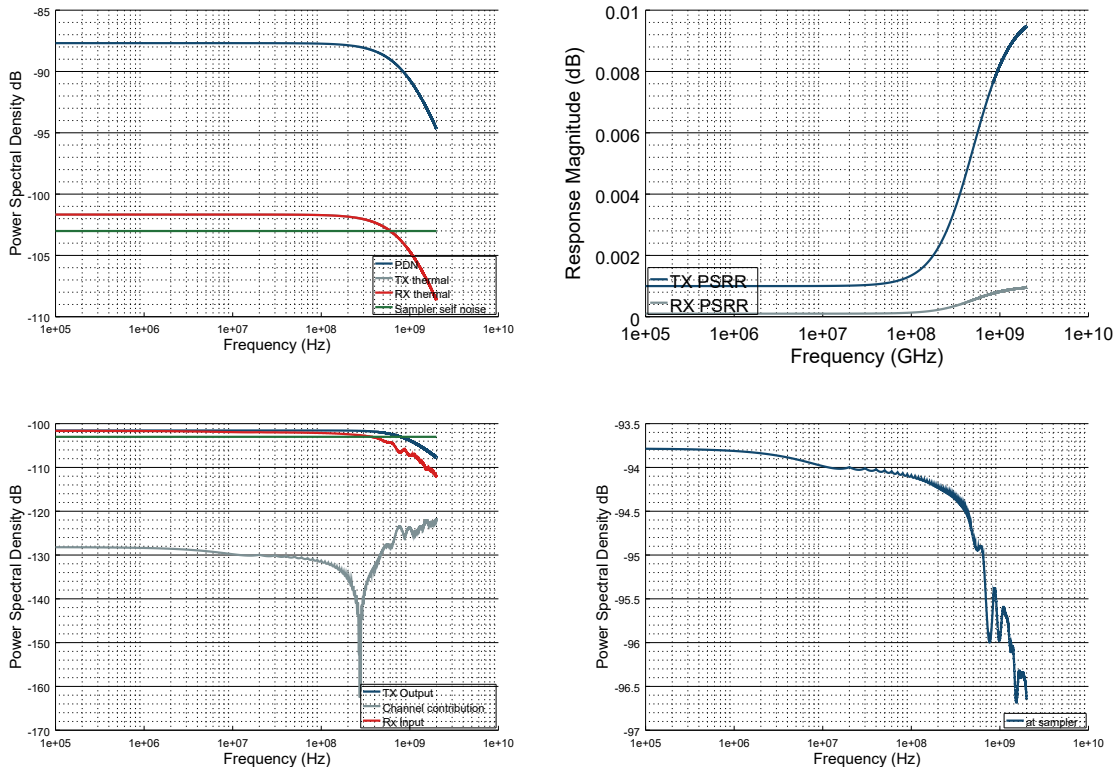

Figure 5.17: Random voltage noise sources in transmitter, channel and receiver and as they may be commonly present in the system and finally appear at the receiver sampler input.

5.3.6 Clock data recovery and sampler phase noise estimation

Most of the clock data recovery parameters can be derived fairly easily (see section 2.2.6) or have to be constrained by design space evaluation efforts directly. The challenge for all digital CDRs mainly lies in estimating the phase transfer function and thus the gain K_{PD} of the bang-bang phase detector (BB-PD). There have been numerous papers on the subject of modeling clock data recovery circuits. We follow the approach taken in [58] for all digital clock data recovery circuits and complement it with information found in [31]. This includes phase and voltage noise modeling as well considering the effects of sampler metastability and intrinsic noise on phase detector gain.

To highlight the difficulties and the special details of the approach taken here, some of the arguments in the respective papers are repeated in the following. To derive the phase detector gain of a BB-PD, we note that the average output of a sampler (comparator) with a DC input voltage level of V_{DC} and an input voltage noise level σ_{V} can be derived to (compare to section 4.5 on intrinsic sampler noise properties)

$$E[d_n](x) = 1 - 2 \cdot \mathrm{erfc}(x) \tag{5.12}$$

where $E[\cdot](x)$ is the expectation value of the digital phase detector output as a function of the continuous variable $x = \frac{V_{\mathrm{DC}}}{\sqrt{2}\sigma_{\mathrm{V}}}$ while erfc is the complementary error function as defined in section 2.1. It is a nonlinear function which could be used to implement the large signal real number model of a phase detector directly. In order to use it within the context of the linear phase domain model presented in section 2.2.6, it needs to be linearized, too. Quite actually, it is the only subcomponent of the CDR with this requirement and thus constitutes the main source of error in the modeling process. As long as the phase and voltage perturbations $\delta\phi$ and σ_{V} remain fairly small (which should be the case for a decently dimensioned serial link), the approximation describes the system well. If no further sampler nonidealities are assumed (as in [58]), the linearization of equation 5.12 around small perturbations will give

$$E[d_n]|_{\frac{V_{\mathrm{DC}}}{\sigma_{\mathrm{V}}} \approx 0} = \frac{V_{\mathrm{DC}}\sqrt{2}}{\sigma_{\mathrm{V}}\sqrt{\pi}} \tag{5.13}$$

In order to arrive at an estimation for K_{PD} which is the slope of the phase transfer function around small phase errors ϕ_{e}, the mean expected DC voltage V_{DC} and the voltage noise σ_{V} need to be related to the timing properties of the input signal. Since here, it is again understood that the small voltage errors translate to phase errors via the signal slope $\frac{dV}{dt}$ at the signal transitions (more precisely, at the sampling points of the edge samplers which should be in the signal crossing points vicinity) we have $\sigma_{\mathrm{V}} = \frac{dV}{dt} \cdot \sigma_{\mathrm{j}}$. The mean DC voltage at the input of the edge sampler is estimated to $V_{\mathrm{DC}} = \delta\phi \cdot \frac{dV}{dt}$ in [58]. Obviously, this assumes fairly homogeneous signal transitions, even though data patterns being sent

will produce ISI and consequently different signal slopes. Also, this assumes the sampler to be equally sensitive to changes at the input irrespective of the phase error magnitude (which is equivalent to a static sampler sensitivity function). Finally, it asserts that dc sensitivity does not depend on the previous decision made by the sampler (i.e. there is no metastability). Instead, [58] estimates a well equalized signal at the edge sampler input to have an amplitude of $2A$ and a slope $\frac{dV}{dt} = A/2$ (in units of $\frac{\mathrm{V}}{\mathrm{rad}}$) which is equivalent to $2A/T$ in units of $\frac{\mathrm{V}}{\mathrm{s}}$. More simply put, this assumes full height of the eye at the center of the UI which may not at all be the case. The height of course depends on the success of the system equalization efforts which are again an integral part of the OCM budgeting procedure. In general, the resulting deterministic jitter PDF will be heavily dependent on the channel and the equalization properties of FIR and CTLE and will not at all adhere to a Gaussian distribution. It is customary and also done in [58] to perform a Gaussian fit to the deterministic PDF in order to derive a reasonable approximation for σ_j. With these assumptions, the phase detector gain can be estimated to

$$K_\mathrm{PD} = \frac{1}{\sigma_j \sqrt{2\pi}} \tag{5.14}$$

in units of rad^{-1}. If the intrinsic offsets of data and edge samplers are not compensated by calibration, this will lead to a dead-band in the phase transfer function (compare figure 5.18 from [58]). The CDR will then wander around the actual target sampling point since there is no meaningful phase detector gain for small phase errors. Since the samplers are offset calibrated within the openMGT OCM budgeting procedure, however, we can safely assume this error to be very small and consider it absorbed by the usual CDR clock dithering (see below). When looking at the sampling point distribution which can be recorded for smaller periods of time during transient simulations, one can verify that this assumption actually is justified for a given implementation.

Yet another, numeric way to derive K_PD is to initially model the phase detector with a step transfer function and take the ensemble average over jitter present at the input to arrive at the expectation value for the output $E[d_n]$ [31]. This can be done by convolving the jitter PDF with the ideal step function. Numerically, this is the same as arriving at equation 5.12 directly - a very interesting circumstance, since it opens a path for integrating the deterministic components of jitter at the sampler input (such as ISI) in the analysis presented here. In addition to phase noise present at the (edge) sampler input, we have to consider the samplers intrinsic property of metastability. It has been suggested [31] to derive the malicious impact of metastability on K_PD by transient (all-digital) simulations with pseudo random input sequences. On the other hand, metastability is a deterministic property of the sampler and is well characterized by the ISF of the sampler (see section 4.5. We will therefore suggest an alternate path further below for deriving an approximate value of K_PD.

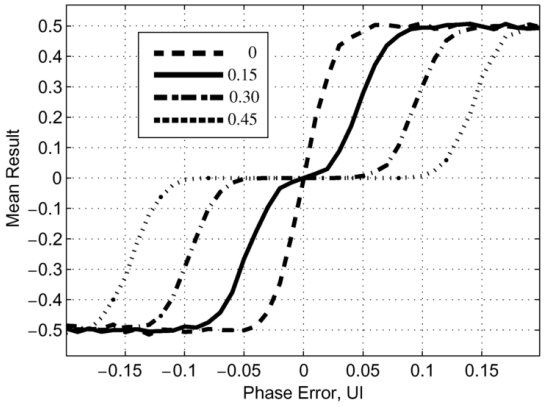

Figure 5.18: Phase detector transfer characteristics under the influence of varying data to edge sampler offset mismatches (from [58]). This problem can be alleviated by proper sampler calibration which is an essential part of the OCM budgeting procedure.

Finally, there are two sources of intrinsic sampler noise that also need to be taken into account here. The first is the broadband intrisic noise of the sampler itself which is referred back to an equivalent input voltage noise and can in turn be mapped to an input phase noise. This noise source is uncorrelated with any noise appearing on the sampler input which allows its resulting noise PDF to be directly convoluted with the jitter PDF at the input. The second source is the so-called *self noise*. The BB-PD produces a binary decision bit for every clock signal transition and an early/late decision for every signal transition. The quantization noise is therefore $\sqrt{2}$ and is white and broadband in nature. The consequence of this noise source is a dithering of the CDR position around its ideal position which gets more severe when CDR bandwidth is not limited [58]. This behavior can be modeled by refering the self noise of the phase detector back to its input, too. It can be shown that the noise source has an equivalent RMS magnitude of $\sqrt{\pi}\sigma_{\mathrm{j}}$. Assuming again this source of noise is uncorrelated with the rest, its PDF can again be convoluted with the input phase noise PDF.

Now that the ideas and problems central to the acquisition of a good estimate of K_{pd} have been presented, the custom OCM budgeting approach can be outlined. The idea is based on the following observations. For one, the phase detector and CDR system performance are affected much differently by deterministic, mainly ISI induced jitter and by random, noise induced jitter. The former is spectrally located at very high frequencies (after all, ISI depends on the specifc bit sequence sent) and can therefore not be rejected/tracked by the CDR. However, its peak to peak value usually is much greater than that of the random noise contribution while at the same time it will not adhere to a Gaussian distribution. However, it is the specific nature of this very distribution which determines the actual

shape of the phase detector transfer function. On the other hand the random noise sources make a much smaller contribution to overall phase gain degradation but do have a lot of spectral content within the CDRs tracking range. Parts of it may therefore be rejected by the CDR. Also, the random noise sources are unbounded. For very low BER operation it is equally important to achieve a good equalization as it is to lower the noise levels at every possible location. All of these preliminary thoughts lead to the following sequence of evaluation.

First, the histogram of the deterministic jitter is determined. The construction of the statistical, deterministic eye of the system channel by performing the PDA analysis takes care of this step - the PDF at the voltage threshold $P_0(t) = P_{\mathrm{eye}}(0, t)$ can then be used for this purpose. The random jitter and voltage noise sources appearing at the input of the sampler have also been found in the previous step. They are available as phase noise power and voltage power spectral density. This information will be needed twice. First, to calculate the overall phase noise level at the phase detector and therefore the linearized CDR loop characteristics. In this first case, the information contained in the spectral densities is simply reduced to the resulting RMS noise level and associated with the appropriate Gaussian PDF. And second, once the linearized model is obtained, the CDR transfer function can be calculated and used to color the spectral noise densities at the inputs to finally arrive at a residual phase noise level and calculate its detrimental effect on the deterministic eye width.

Therefore, instead of using the simple deterministic transition PDF at the threshold as suggested above, the deterministic eye is first convoluted with the voltage noise PDF which also has the effect of averaging the potentially sparsely populated matrix elements around the transition. A resulting deterministic, ISI induced PDF with random voltage noise contribution is shown in figure 5.19 as it results from the deterministic eye diagram and the sampler input voltage noise as given in figures 5.13 and 5.17 respectively.

Now, the random contributions to jitter as seen at the CDR input also have to be taken into account. The previously obtained time interval error PDF is therefore convoluted with the PDFs of the statistical receiver input phase noise spectral density (potentially degraded by channel jitter amplification) as well as by the receiver reference clock jitter PDF and the phase interpolator dither PDF whose magnitude is given by the phase interpolator resolution and the current data rate via

$$\phi_{\mathrm{PI,mag}} = K_{\mathrm{DPC}} \cdot T$$

in case of a digital dither with a magnitude of a least significant PI bit. The resulting, complete time interval error PDF can be seen in figure 5.20.

It is now asserted here, that the PD characteristic of metastability can be obtained by convolution of the ideal, step like phase detector characteristic (blue curve in figure 5.21)

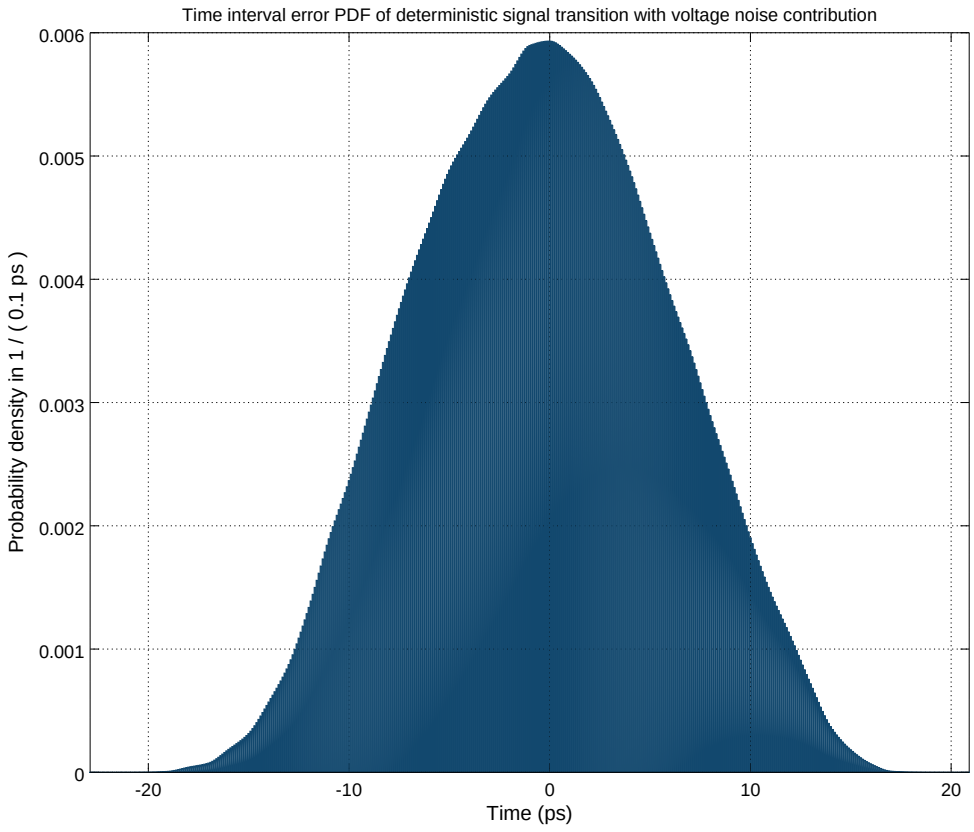

Figure 5.19: Preliminary probability density function as it may appear at the sampler inputs of a CDR circuit.

with the complete TIE PDF (grey curve) and the impulse sensitivity function of the sampler, resulting in the red curve of figure 5.21. As described in section 4.5 the convolution of the ISF with a step function will lead to a value that, if compared to the samplers dc sensitivity threshold, will determine if a digital zero or one is resolved. The sensitivity parameter affects the decision depending on the previously resolved bit. If the process of comparison (thus, digital resolution) is omitted, we instead obtain the expectation value $E[d_n]$ of the ensemble average of all possible data pattern combinations with their signal transitions all located at the specified phase offset to the sampling clock instant. Just this time, the expectation value is not resulting from phase noise present at the phase detectors input, but from its intrinsic, metastable behavior (which by itself is of course deterministic) and the random resolved states of the sampler which are due to the randomness of the received data. Strictly speaking, this assumption only holds if logic zeros and ones are equally probable within the data stream (an assumption already made in the context of constructing the PDA statistical eye). We then take the derivative at a phase error of zero. The process of obtaining K_{pd} involves the consideration of all input noise sources in their unfiltered

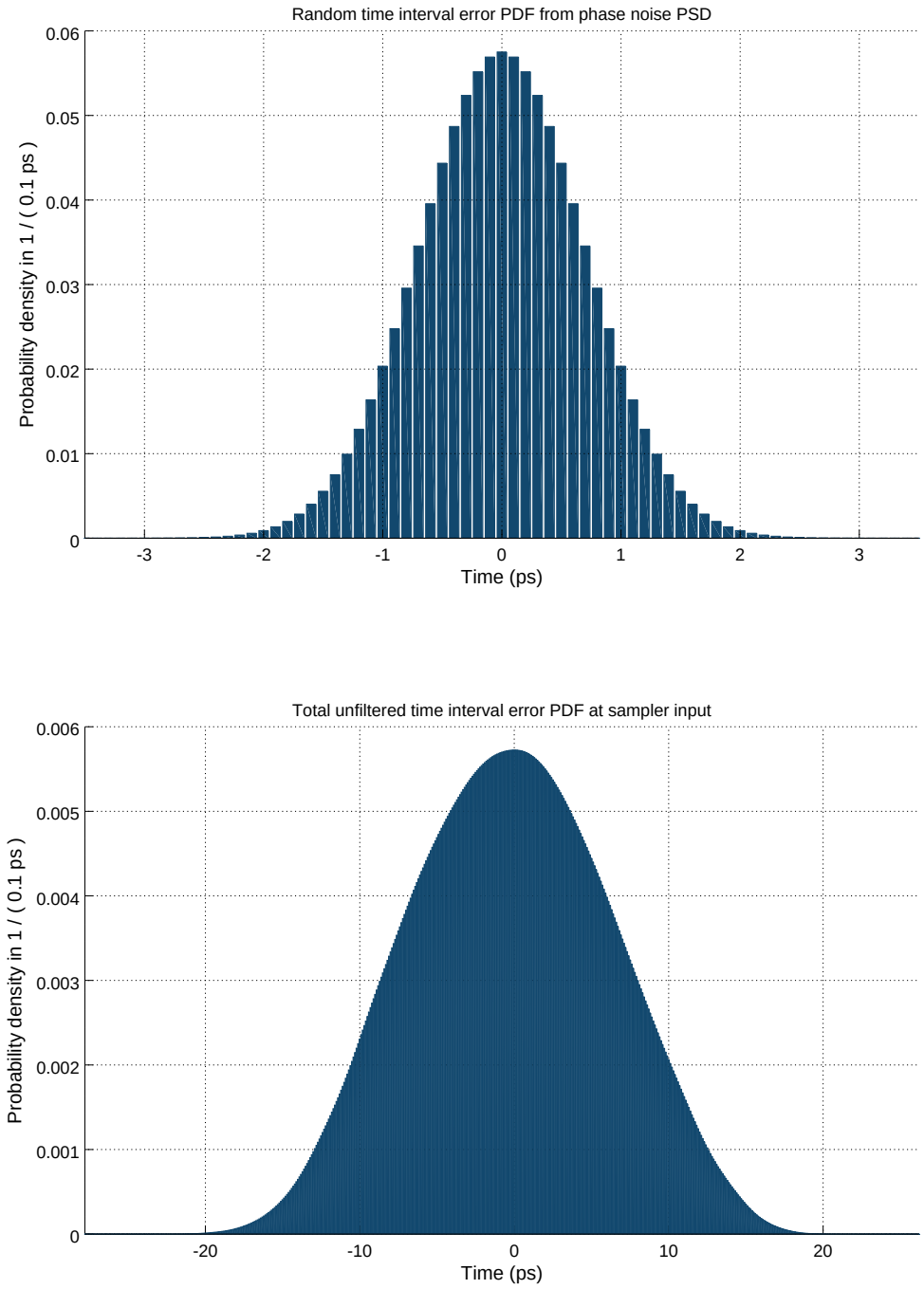

Figure 5.20: Complete probability density function as it may appear at the sampler inputs of a CDR circuit.

form. This is necessary because unlike the resulting, recovered clock $\phi_{\text{out}} = \phi_{\text{samp}}$ the input data stream does not at all undergo any type of filtering process. It is however this full phase noise, or rather its resulting RMS jitter, which will determine the slope of the phase transfer function of the BB-PD and thus the CDR recovery loop gain.

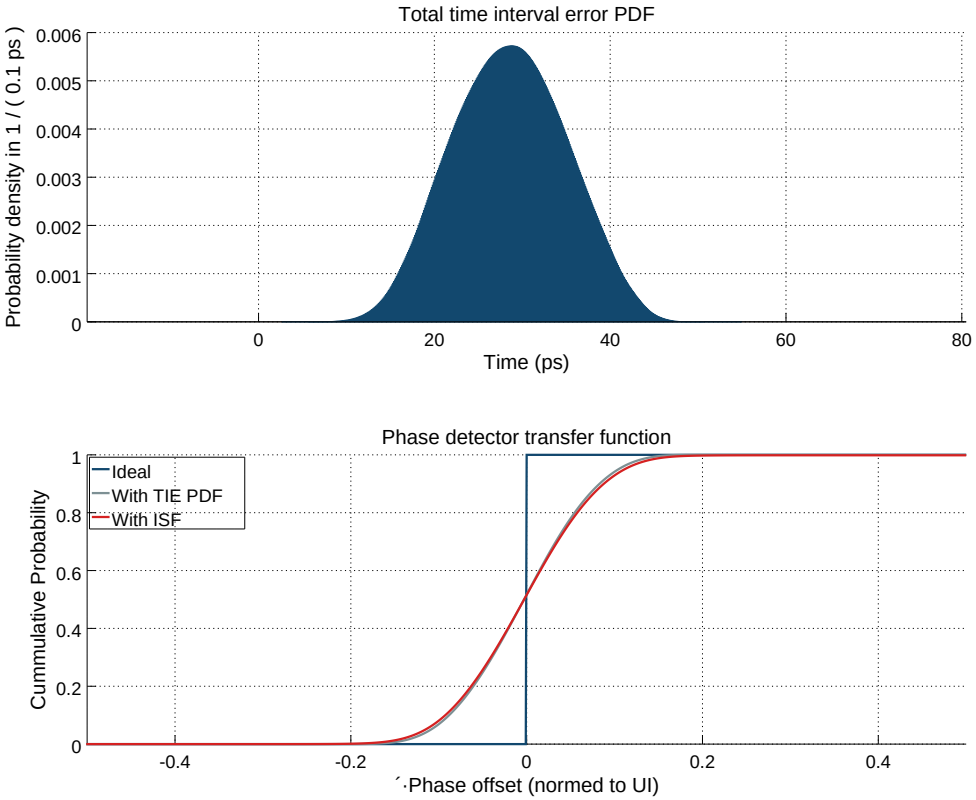

Figure 5.21: Graphical representation of phase detector calculation via convolution with the TIE PDF and sampler ISF

For construction of the linearized CDR phase transfer function, Kunderts approach on applying phase noise spectral densities to the CDR system is combined with the digital loop filter description by Sonntag while using the custom approach to determine effects of metastability and input phase noise PSD. The loop gain and jitter transfer function are defined as in section 2.2.6. As can be seen from figure 2.13 in section 2.2.6, the phase error function can be derived to

$$\phi_{\text{err}}(s) = \frac{\phi_{\text{in}}(s) - \phi_{\text{rx}}(s)}{1 + H_{\text{CDR,loop}}(s)}$$

It is the phase error we are interested in since it is this very property of the receiver

system that gives timing jitter on the input data stream its meaning - it is the quantity, that the BB-PD will eventually convert to an average digital representation and it is this representation which will be filtered and used as an estimate of the phase of the incoming clock embedded in the data stream. Now though, this phase error actually is affected by the CDR loop filtering properties. The input phase noise and the RX reference clock phase noise power spectral densities are the known quantities and the resulting phase error power spectral density can consequently be derived to

$$S_{\phi_{\mathrm{err}}\phi_{\mathrm{err}}}(\omega) = \frac{S_{\phi_{\mathrm{in}}\phi_{\mathrm{in}}}(\omega) + S_{\phi_{\mathrm{rx}}\phi_{\mathrm{rx}}}(\omega)}{1 + H^2_{\mathrm{CDR,loop}}(\omega)}$$

A phase noise spectral density plot as resulting from this filtering process is shown in figure 5.22 where the CDR can be seen to effectively remove the spectral content up to frequencies set by its own loop bandwidth. It is shown without PI dither contribution. To its top, the resulting time interval error PDF as required to construct the final, statistical eye for system metric evaluation is also presented.

While not impossible to estimate, $S_{\phi_{\mathrm{rx}}\phi_{\mathrm{rx}}}(\omega)$ is tough to thoroughly model. Apart from the initial phase noise of the PLL, the entire clock distribution tree with the rather complicated chain of clock buffers and shapers in the DPC would need to be modeled and its model parameters estimated. The framework would generally support this procedure as described in section 2.2.5. On the other hand, as a starting point in design space explorations or for reasons of simplicity, an upper bound estimation may also be used. Once the schematic based implementation has reached a mature state, a PXF and periodic noise analysis can then be used to derive the then rather exact form of the receiver reference clock phase noise.

As explained above, the input data stream phase noise has far more contributors to its overall shape. The previous sections already explained the shaping process of the PLL/TX output phase noise by the channel and jitter amplification in the presence of reflections and equalization. It must be noted here, that estimating the phase noise at the input of the phase detector should not include the equalization effects of the DFE in cases when there is no feedback summation stage feeding the inputs of the edge samplers. Whether it is worth the additional effort of feedback-equalizing a signal transition, which kind of clock timing would need to be used for this and also which consequences this may have for the overall CDR loop dynamics are all questions beyond the scope of this text and open for exploration with the framework presented here.

Finally, digital implementations of clock data recovery circuits do not simply implement a single value for the proportional and integral gain. They can and should be adjustable by digital control vectors. If such vectors are present, the numeric optimization routines of Octave and its OCM extension can be used to advise on optimal settings for these vectors with respect to the overall residual phase noise level for clock recovery.

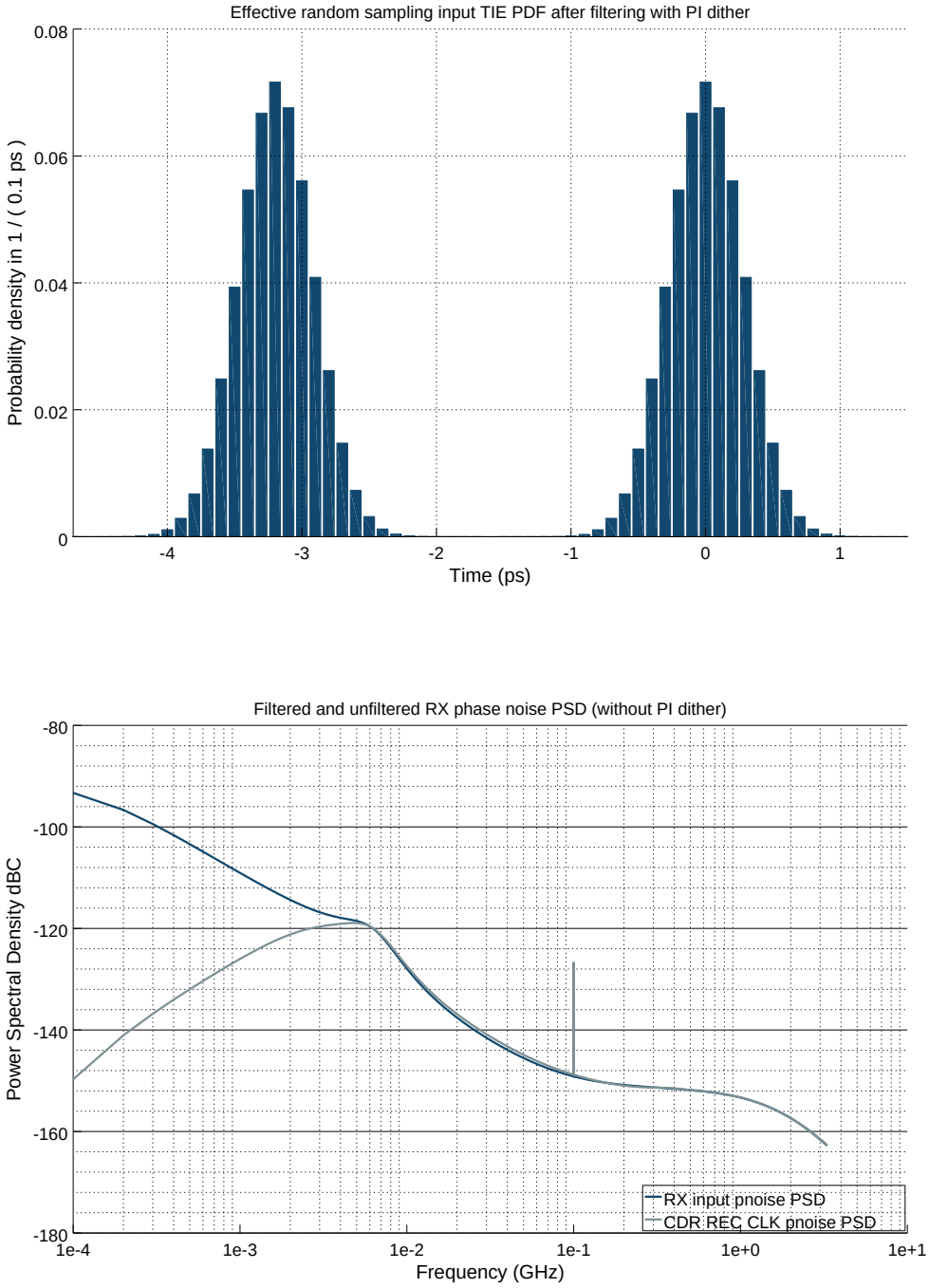

Figure 5.22: TIE PDF as well as unfiltered and filtered sampler input phase noise PSDs resulting from analysis procedure presented here

5.3.7 Final statistical eye compilation and metric extraction

The final, statistical eye is calculated by convoluting the phase noise PSD of the previous step "horizontally" with the deterministic PDA eye. Also, the sampler aperture is considered in the same way which in conjunction with the "vertical" convolution of the sampler input noise as previously calculated and the subtraction of the sampler dc sensitivity culminates in the so-called *post aperture eye*.

An example of the resulting eye is shown in figure 5.23 below in logartihmic color scale.

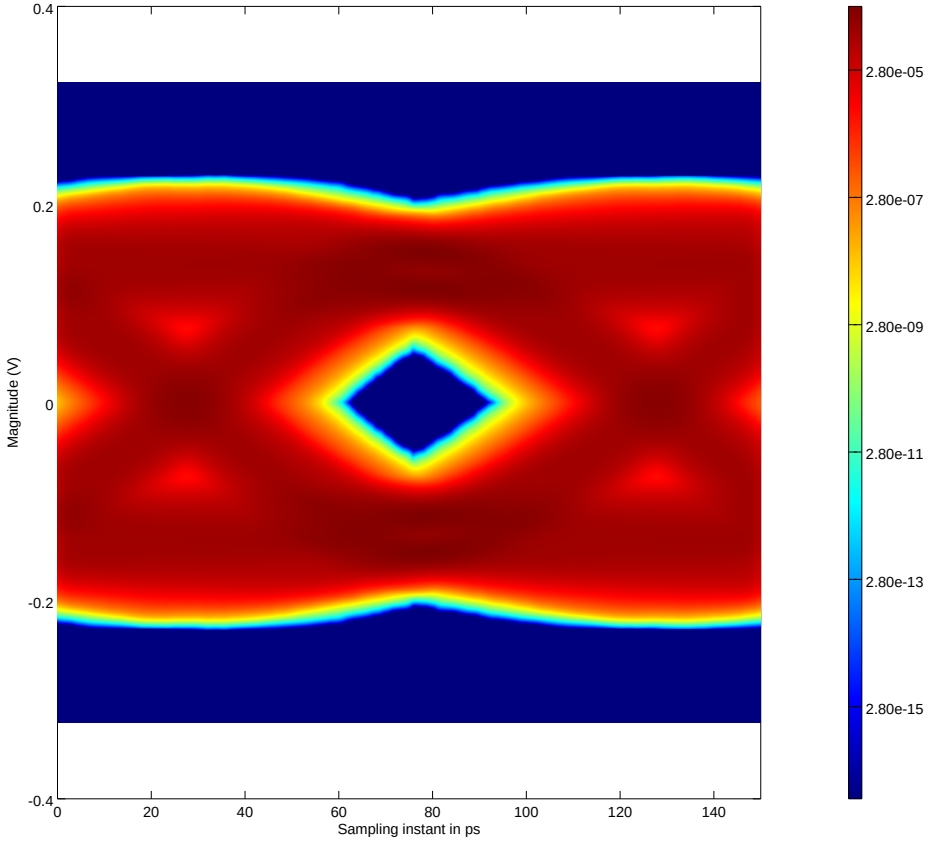

Figure 5.23: Final statistical eye diagram as reported by a budgeting/analysis procedure run. In this case, a target bit error rate of 10^{-15} can well be achieved.

Figure 5.24 shows the conture plot of the eye diagram of figure 5.23. While the eye diagram is a probability density distribution, the contour plot is based on its cumulative sum as required by the definition of the BER quantitiy. The result are isoBER lines which enclose the area where the BER is equal or less than a specified value. More common is the representation along one of the eyes dimensions, usually along the time axis at the

threshold voltage. This plot is shown in figure 5.25 and is called the *bathtub* curve. From it, both quality of equalization and severity of noise sources can directly be seen. The one by noting that deterministic contributions to eye closure occur at low probabilities and have more gentle slopes while the BER towards the center of the eye and for low probabilities is goverend by the derivative of the tail of a Gaussian PDF. Its slope thus is directly proportional to the standard deviation of the underlying total random noise PDF.

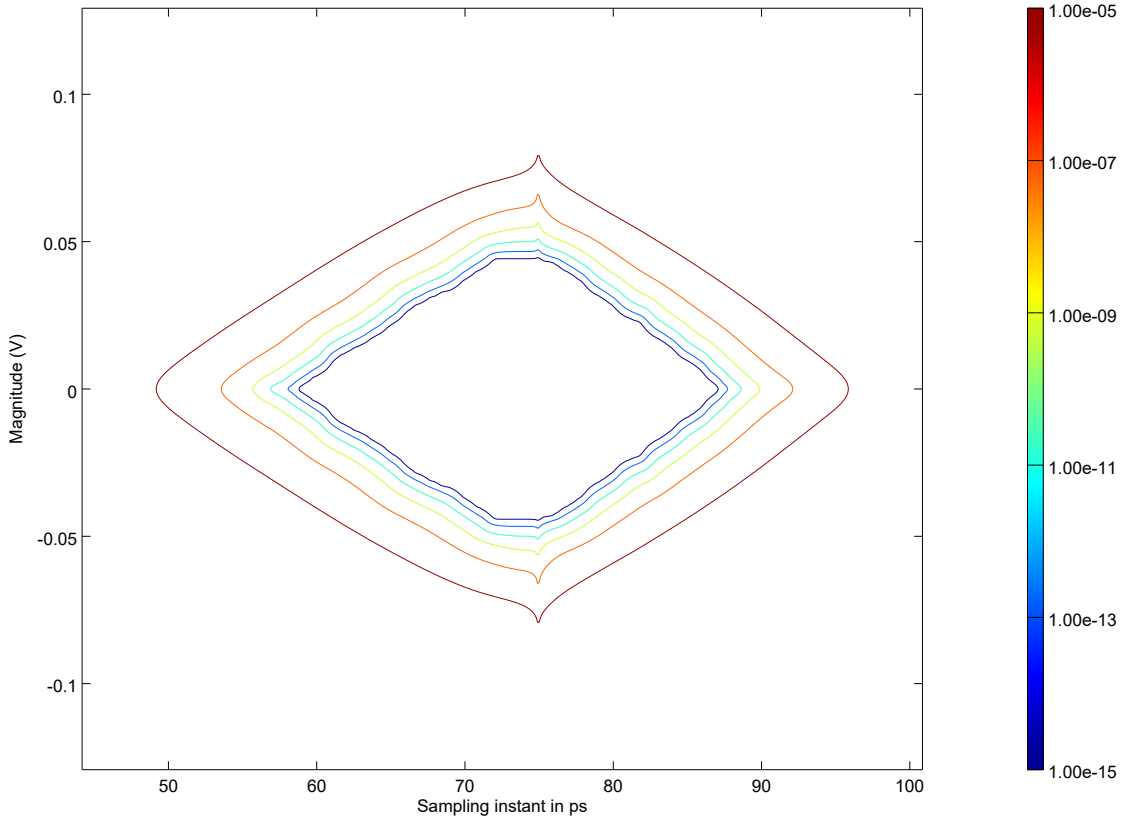

Figure 5.24: Isometric bit error rate lines of the eye diagram in figure 5.23 (note the time scale difference with respect to the statistical eye diagram for better visibility)

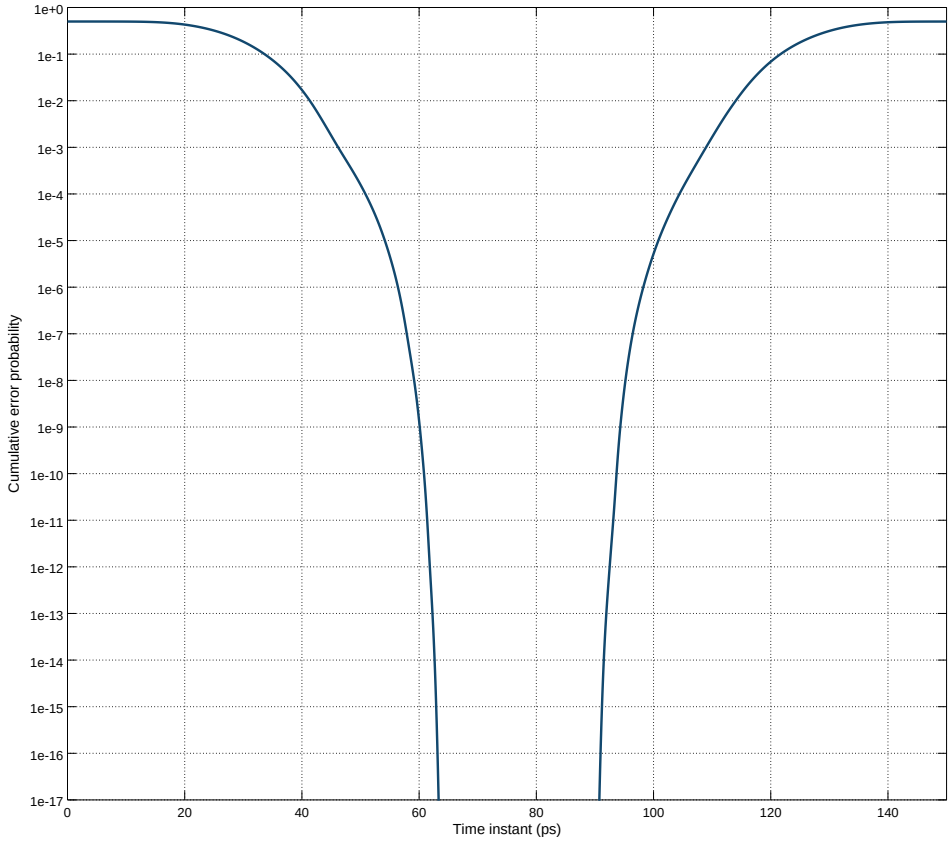

Figure 5.25: Bathtub curve of the bit error rate at the decision threshold for the statistical eye diagram of figure 5.23

6 Design evaluation

The budgeting and analysis framework developed in the previous chapters was used to develop a serializer system for data rates ranging from 2.5 to 20 Gbit/s. This was done as part of a team effort during conception of this thesis. This chapter presents the general architecture of the design and highlights some interesting analysis aspects.

6.1 A quarter rate serializer design

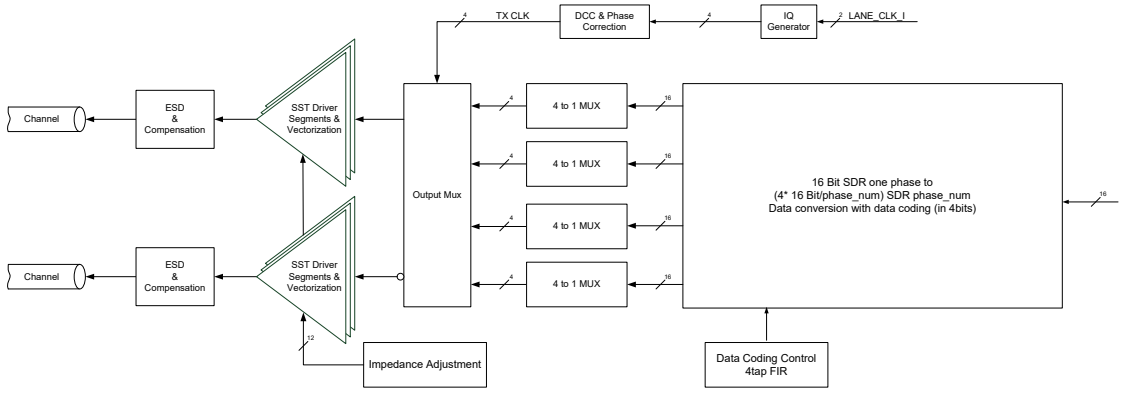

Figure 6.1: Quarter rate transmitter architecture, after [38]

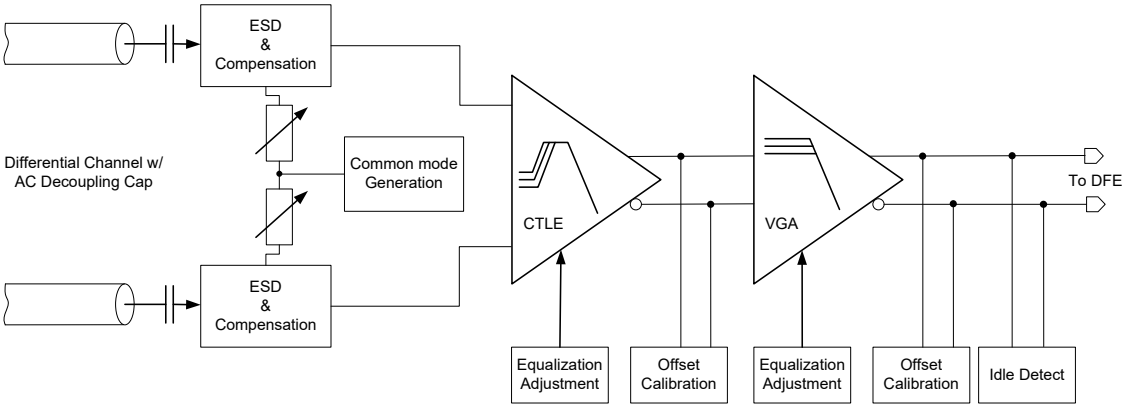

Figure 6.2: Receiver analog frontend

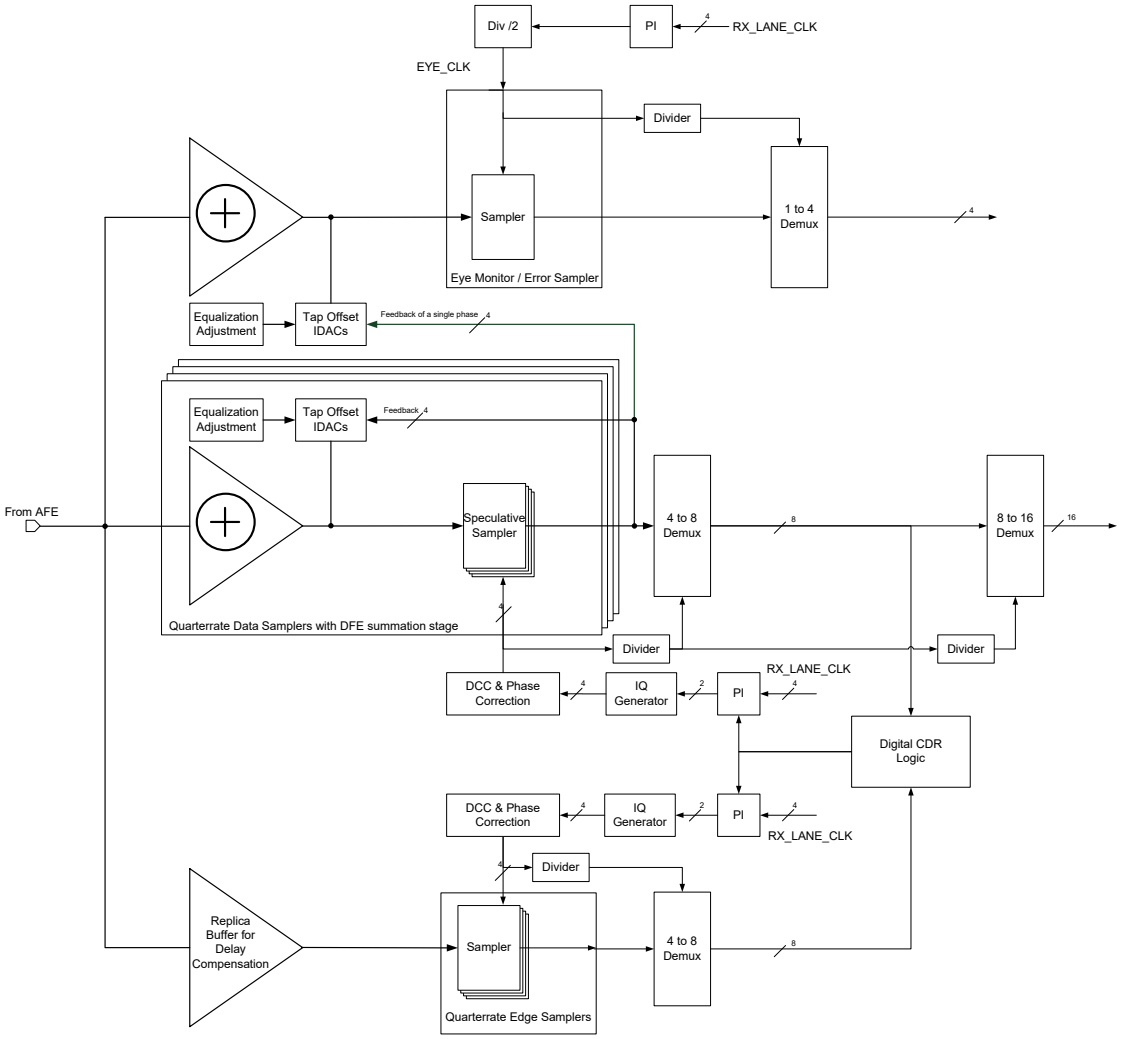

Figure 6.3: Quarter rate receiver with DFE, clocking scheme and its digital architecture, after [38]

Figures 6.1 through 6.4 provide an overview of the serializer transmitter, receiver analog frontend, general receiver structure and the structure of a speculative sampler respectively.

The transmitter is based on a quarter rate architecture which is implemented in HDL to the largest degree. The only exceptions to this are the ESD compensation, the SST driver segments, the clocking tree with duty cycle correction (DCC) and phase adaption as well as the output multiplexer. The output driver actually accepts four input streams of data, each to specify a given tap for the four tap FIR implementation. Its output impedance can be tuned. The proper line coding is accomplished by the HDL code which accepts 16 bit at its parallel interface per core clock cycle (single data rate, SDR), recomputes the data according to user defined tap adjustments and distributes the data stream to the four phase channels accordingly. Much more information on the general architecture and the possibility to support other coding schemes than FIR weighted NRZ signaling are presented in Ref [56].

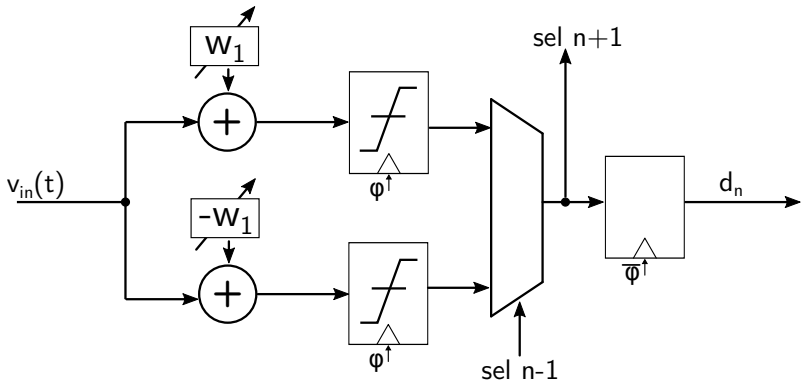

Figure 6.4: Speculative sampler in the data path

The receiver analog frontend is an entirely analog design, apart from its offset calibration and equalization adjustment vectors. The transmission line is terminated with an adjustable load impedance. This is to compensate process variations and to support a small range of waveguide impedances in the range of $40 - 60\,\Omega$. Also, an adjustable common mode generator is included to set the appropriate operating point for the first amplification stage. This is especially important in cases where the transmission line includes AC coupling capacitors such as is typically the case in backplane and network applications. The ESD compensation is required to account for the malicious, capacitive loading of the diode structures. It is basically a so-called T-Coil compensation [35] which makes the combined structure of inductance and diode capacitance appear like a part of the impedance defined transmission channel. This minimizes the back-reflection of signal power into the channel at higher frequencies. As a consequence the bandwidth of the forward transfer function from receiver input to the first amplifier stage is also extended. The CTLE possesses adjustment vectors for resistive and capacitive degeneration which allows to control both the zero and the first pole of the (idealized) three pole transfer function. Since the CTLE and the subsequent variable gain amplifier (VGA) provide most of the gain in the amplification chain leading to the receiver samplers, both are equipped with offset calibration circuitry. The VGA quite actually is also an equalizer albeit with a smaller range of adjustment options and larger degeneration capacitances. Its major purpose is to keep the voltage swing at lower frequencies limited to a reasonable range. This avoids strong nonlinear effects in the succeeding stages and large overdrive voltages at the receiver sampler. Additionally, the capacitive degeneration allows to avoid attenuation at higher frequencies which have just previously been elevated by the CTLE. All the components within the analog frontend have a real number model representation with an adjustable set of parameters such as their maximum DC gain, degeneration resistor and capacitor ranges, input and output common modes, intrinsic, randomly generated offsets and, of course, resulting frequency transfer functions.

The receiver architecture, too, is a quarter rate design. It features a speculative decision feedback equalization circuit [30] *without* adjustment of the phase detector (edge) path. The speculative samplers operate on input data whose detection threshold is shifted in accordance with the settings of the first DFE tap in the two opposing directions. The bit previously resolved, once known, then decides which of the two speculative decisions is actually latched and forwarded to the digital domain (see figure 6.4).

Additionally, for debugging and automatic in system equalization, there is an eye monitor path with a single speculative sampler (actually two then) of adjustable, relative sampling position and voltage offset. The DFE introduces a further amplifier into the analog signal chain leading up to the fourteen samplers - the summation buffer. It is a resistive amplification stage with four current feedback taps (as the first tap is implemented by speculation). The sampler clocks are provided by an advanced and involved clocking scheme based on phase interpolator circuits [38]. The receiver clock from the reference PLL actually possess twice the frequency required. Once it is shifted by the phase interpolators in accordance to the vectors provided by the all digital CDR circuit, the IQ generator blocks create phase matched and duty cycle corrected half rate versions which can then be used by the sampler circuits. Dividers are then used to derive the clocks needed for demultiplexing the signal for the lower frequency, parallel interface. Due to technology constraints, the CDR, too, needs to operate at a lower data rate. It must be chosen such that timing can still be met with semicustom implementation flows. Choosing the data rate as high as possible on the other hand allows to increase the loop bandwidth more easily (compare equation 2.12 of chapter 2) and therefore improve the jitter rejection of the resulting system. Here, the achievable maximum for the targeted data rate of 20 Gbit/s was the clock domain running at an eighth of the baud rate and thus at 2.5 GHz. For many more details on the actual architecture the reader is referred to [38].

For the subsequent analyses in accordance with the procedure presented in the previous chapters, the budgeting parameters chosen are given in Appendix B unless otherwise noted in the specific cases below.

6.2 Analysis of 10 Gigabit operation

The serializer is analyzed for compliance with the *channel stressor* of the 10 Gigabit Ethernet specification [17]. The focus here lies on the real number modelling layer of the serializer, not the actual schematic implementation. The target bit error rate is over-constrained to 1e-13 to account for this circumstance. The simulations are carried out with a time resolution of 5 ps except for the sampler with a time resolution of 1 ps, whose Cauchy-Lorenz model is constrained to an aperture of 4 ps, a sensitivity of 0.5 mV, a self noise level of 1.5 mV_{rms} and an aperture delay of 28 ps. Random offsets as well as duty cycle distortion as a result of clock buffer skews omitted in this and the analyses to follow.

The framework is used to compare the effect of equalization on system performance first. For each equalization setting, table 6.1 lists the most central results of the budgeting procedure. Also, the system transfer function and resulting single bit responses are shown for all settings. The analyzed settings are

- #1 - the use of a CTLE preset in conjunction with automatic FIR and DFE adaption
- #2 - automatic adaption of the FIR with deactivated DFE and only globally attenuating CTLE and VGA
- #3 - automatic adaption of the FIR with deactivated DFE and CTLE and VGA preset of #1
- #4 - automatic adaption of the DFE with deactivated FIR and only globally attenuating CTLE and VGA
- #5 - automatic adaption of the DFE with deactivated FIR and CTLE and VGA preset of #1

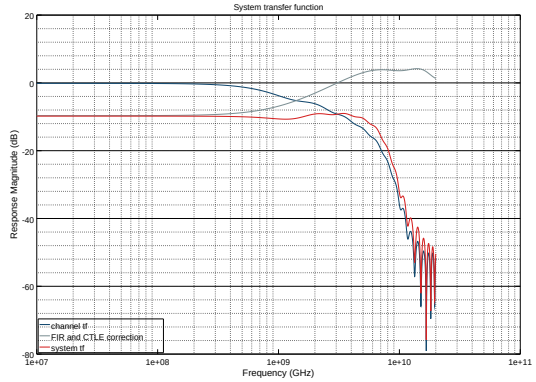

Figure 6.5: System frequency response (without DFE) of equalization setting #1

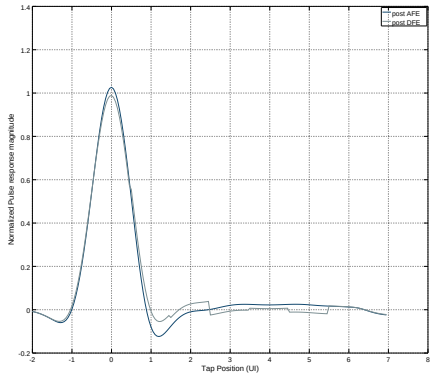

Figure 6.6: Single bit response (with and without DFE) of equalization setting #1

For the first case, the frequency response of the channel is given in figure 6.5. As can be seen, the system transfer function is flat up to about 5 GHz. The CTLE has a zero at 1.05 GHz and its pole located at around 6.6 GHz as roughly expected from the bare channel transfer function. The SS-LMS adaption procedure for the four tap FIR converges to a setting of $(-0.07, 0.93, 0, 0)$. The main cursor is therefore only slightly reduced in favor of removing precursor ISI in the final SBR. The postcursor taps are not used at all since the CTLE already takes care of this aspect. The remainder of residual ISI in the post analog receiver frontend single bit response (blue curve in figure 6.6) is then taken care of by the

DFE up the to fifth postcursor. Its taps are set to $(1, 1, -1, -1, -2)$ where the value range of the individually weighted DFE tap current DACs is $[-15, 15]$ and the relative weighting between the DACs is $150 : 100 : 75 : 50 : 25$ from first to fifth posttap. As can be seen from the DFE tap vector and the light gray, DFE corrected SBR in figure 6.6, the DFE actually needs to remove some of the negative ISI in the first post cursor introduced by the slightly overequalizing CTLE. This overequalization can also be seen from the small but noticeable lobe in the system channel transfer function (red curve) of figure 6.5 around 3.5 GHz.

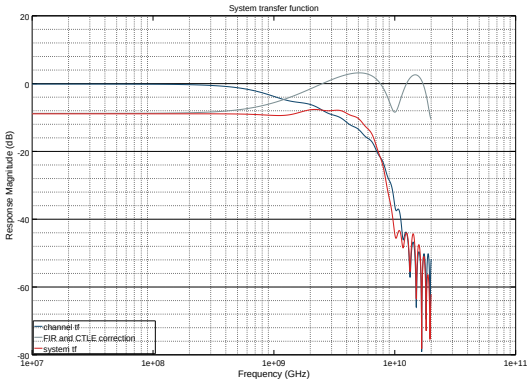

Figure 6.7: System frequency response (without DFE) of equalization setting #2

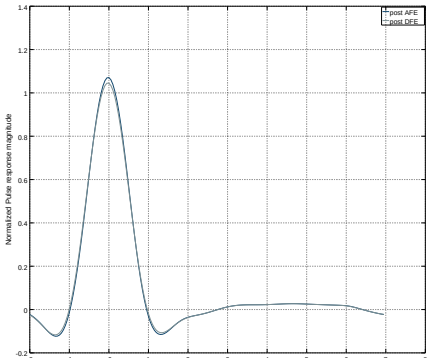

Figure 6.8: Single bit response (with and without DFE) of equalization setting #2

The second equalization setting makes no use of DFE equalization and chooses CTLE and VGA settings such that their frequency response is (almost) flat with a DC gain of 1.46 and a zero at 12 GHz, well beyond the Nyquist frequency of the data rate. It delegates correction of the channel transfer function solely to the FIR and chooses its four taps according to $(-0.08, 0.62, -0.274, -0.02)$. The resulting system transfer function and the single bit response are shown in figures 6.7 and 6.8. While first pre and postcursor are perfectly cancelled, a slight overequalization of the main cursor (larger than unity) can be observed. Also, due to the finite resolution of the TX FIR, the second postcursor is mildly overcorrected. This, however, highlights one of the advantages of the simulation and modelling approach presented here - the deterministic errors resulting from finite DAC or ADC resolutions are captured as deterministic effects.

The third setting adapts the FIR with the CTLE settings as given in the first iteration while still leaving the DFE unused. The four FIR taps converge to $(-0.07, 0.79, 0.09, 0.05)$. Again it can be seen that the two post cursors anti compensate the overequalized setting of the CTLE while the precursor remains at almost the same value as in the cases before.

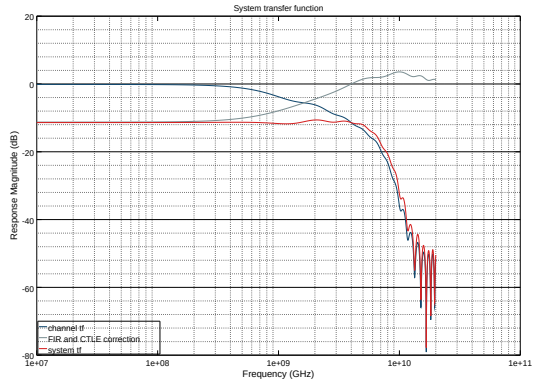

Figure 6.9: System frequency response (without DFE) of equalization setting #3

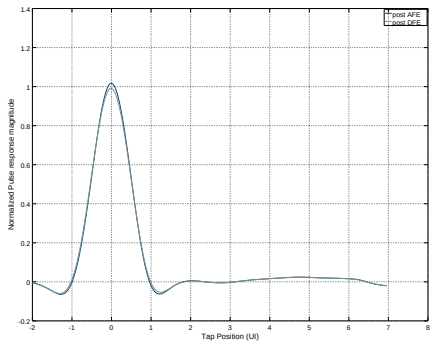

Figure 6.10: Single bit response (with and without DFE) of equalization setting #3

From table 6.1 it can be seen that this effectively trades eye height against an increase in eye width. As can be seen from figure 6.10, the overequalization of the main cursor is removed and the second post cursor manages to nearly zero out the ISI at its location. As a consequence, the system frequency transfer functions flatness is superior to those of the preceding examples.

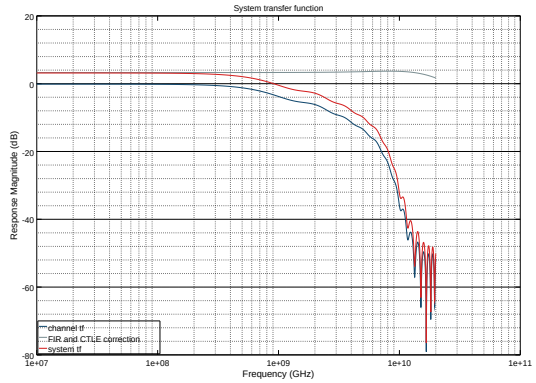

Figure 6.11: System frequency response (without DFE) of equalization setting #4

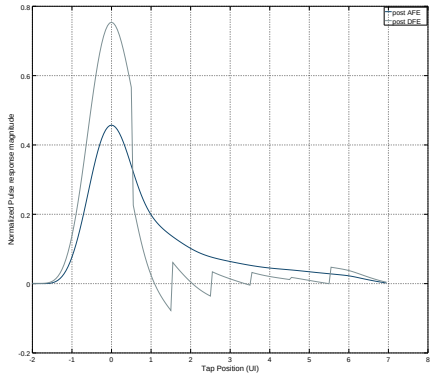

Figure 6.12: Single bit response (with and without DFE) of equalization setting #4

The fourth setting uses neither FIR nor CTLE (apart from a signal gain of 1.46). As a result, the system transfer function without DFE is unchanged in shape, the resulting single bit response after the receiver analog frontend exhibits large magnitudes of ISI which are completely cancelled by the first speculative and second, current DAC based tap. The DFE tap weights which the SS-LMS algorithm converges to are $(-15, -12, -9, -8, -7)$. It is unclear why the last taps do not converge to more negative values as the tuning range

would still provide more headroom and the main cursor of the SBR is not fully equalized (compare figure 6.12 where the main cursor tap is much smaller than unity). The main reason for the main cursor not being fully equalized, however, lies in the fact that the strongest contributors - the first two taps - reach cancellation of ISI at their position which consequently stops the main cursor from being increased much further. From table 6.1 it can clearly be seen how strongly the DFE can increase the signal to noise ratio and the overall signal level at the sampler in this case. On the other hand, since the edge sampler input signals do not undergo any kind of equalization, the resulting phase detector gain is very low. As a result, the CDR bandwidth and the jitter rejection capability strongly decrease which in turn leads to an elevated level of random jitter. The increase in eye height and deterministic width, however, seem to compensate for this shortcoming, although it must be mentioned that the budgeting framework reports the CDR to have a tendency of becoming instable due to its phase margin dropping below sixty degree.

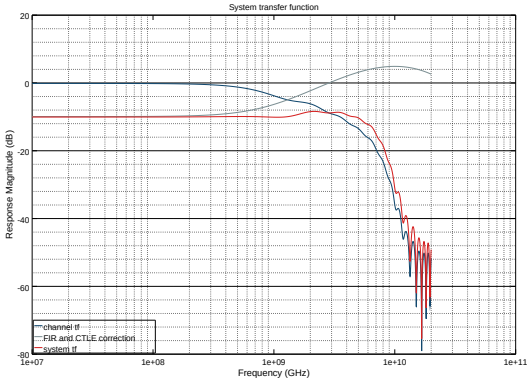

Figure 6.13: System frequency response (without DFE) of equalization setting #5

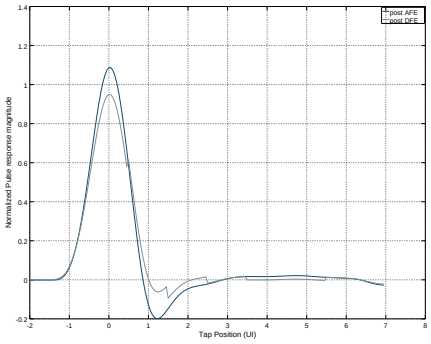

Figure 6.14: Single bit response (with and without DFE) of equalization setting #5

Finally, the fifth iteration activates the CTLE again with the settings from #1 and tunes the DFE taps accordingly. The resulting frequency transfer function and SBR can be seen in figures 6.13 and 6.14. The equalization converges to the DFE tap stengths of $(2, 1, 0, -1, -1)$. Again, the inevitably inferior phase detector gain can be observed (see table 6.1). On the other hand, the results are quite comparable to those of an FIR only adaption apart from the superior eye width. From an equalization point of view, the CTLE is performing the large bulk of the task. The DFE even needs to anti correct the slight over-equalization in the first and second tap for the Ethernet stressor channel.

Table 6.1 gives a summary of the most important budgeting results of the previous examples, where

- $V_{\mathrm{lo/hi}}$ describes the mean high/low level peak to peak,
- $j_{\mathrm{d,pkpk}}$ is the deterministic jitter of peak to peak,
- the CDR phase detector gain is given by K_{PD},
- the CDR bandwidth is described by f_{cdr},
- the post CDR random jitter given by $\sigma_{\mathrm{j,cdr}}$,
- the total noise voltage noise seen at the sampler inputs (including self noise) denoted by σ_{V}

and finally with the resulting eye width e_{w} and eye height e_{h} at the target BER.

EQ	$V_{\mathrm{lo/hi}}$	$j_{\mathrm{d,pkpk}}$	K_{PD}	f_{cdr}	$\sigma_{\mathrm{j,cdr}}$	σ_{V}	e_{w}	e_{h}
#1	±87.1 mV	10 ps	9.15	10.1 MHz	0.216 ps	2.85 mV	57.3 ps	116.9 mV
#2	±93 mV	15 ps	7.35	8.2 MHz	0.24 ps	2.79 mV	49.2 ps	112.1 mV
#3	±73 mV	7 ps	11.6	13.1 MHz	0.19 ps	2.86 mV	59.6 ps	102.6 mV
#4	±212 mV	7 ps	2.24	3.2* MHz	0.38 ps	2.79 mV	59.6 ps	171 mV
#5	±82 mV	10 ps	7.33	8.2 MHz	0.24 ps	2.86 mV	56.5 ps	116.9 mV

Table 6.1: OpenMGT/OCM Analysis results for 10 Gbps operation of the presented serializer architecture

These analyses highlight that the voltage noise at the sampler in this example is solely affected by the choice of equalization in the receiver analog frontend. A large portion of this noise actually is the sampler self noise. The framework could in principle be extended by either a numeric or table based prediction of the frequency dependent noise figure of the analog frontend which would increase the accuracy of the simulations presented here.

Generally, a better equalization before reaching the DFE leads to a better phase detector gain which in turn increases the CDR jitter tracking bandwidth and therefore decreases random jitter contributions which may then increase the eye width drastically due to the Q-scaling effect at low BER if the eye height is not decreased too drastically. The large signal to noise ratio increase by DFE techniques can clearly be observed in the set of examples.

In summary, the serializer architecture in its real number model configuration presented here with the chosen subcomponent metrics shows compliance with the particular physical channel (stressor) of 10G Ethernet, regardless of the chosen equalization approach. If schematic implementations adhere to the so defined metrics (and do not introduce analog problems uncaptured by the modelling process), the serializer system will be suitable for the targeted environment. Quite actually, the system is clearly overconstrained for the given task. The equalization procedures do not have to be used to their full potential. However, since the given architecture is designed to support even higher data rates, this certainly is a good sign.

6.3 Analysis of 20 Gbps operation

It is therefore interesting to see, how the system will perform at higher data rates and with a suitable channel for that operation. For this purpose, the CEI-25G-LR compliant channel from chapter 3 is used (see figure 3.1 or the figures below for its frequency transfer characteristic). The data rate is set to 20 Gbit/s and the simulation time step is decreased from 5 ps to 1 ps to account for the smaller UI of 50 ps. The CTLE is set to have its zero located at 0.48 GHz and its pole at 3.87 GHz. Additionally, the VGA is used as a second CTLE with the zero located at 7.19 GHz and the pole at 8.85 GHz. The total DC attenuation of this second order CTLE lies at −12 dB. The analysis procedure performs an equalization convergence attempt on the FIR and DFE filter taps. This results in an FIR tap setting of $(-0.2, 0.775, 0.01, 0.01)$ and a DFE tap setting of (−5,3,5,5,3). The resulting system transfer function can be seen in figure 6.17. The final statistical eye and its bathtub curve at the decision threshold are shown in figures 6.15 and 6.16. Eye width and height are reported as 6.3 ps and 0.376 mV respectively. Especially the eye height appears to be very small here. It has to be kept in mind, however, that this height accounts both for the random noise sources *and* the imperfections during analog to digital conversion. As long as the value is greater than zero, the chosen architecture and its setting have actually

passed the verification test. The system transfer function shows overequalization for the chosen CTLE and VGA settings which are corrected by the FIR and DFE filter. This suboptimal settings leads to more signal attenuation than necessary. Despite this fact, the post processing procedure reveals the chosen settings to be generally acceptable for 20 Gbit/s across the given channel.

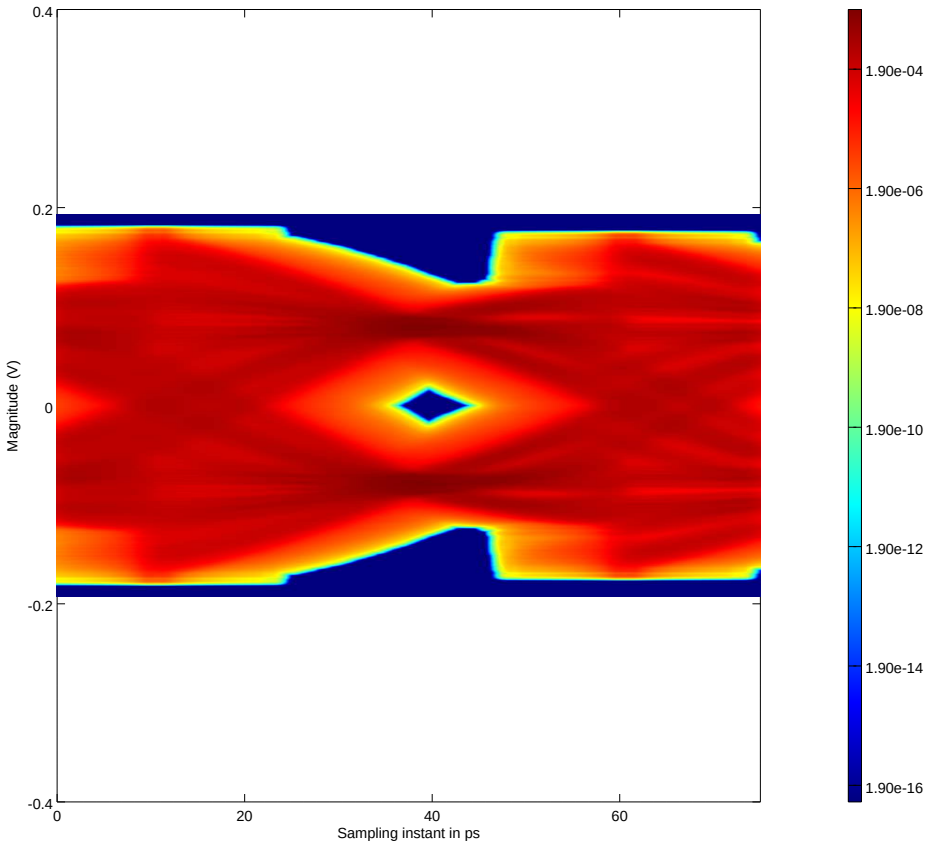

Figure 6.15: Statistical eye diagram for 20 Gbit/s operation the CEI-25G-LR worst case channel of figure 3.1

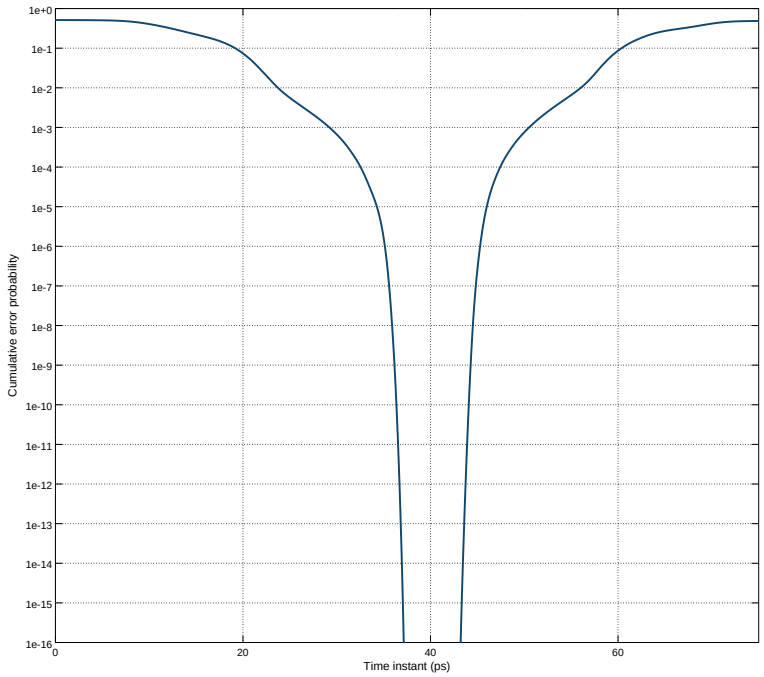

Figure 6.16: Corresponding bathtub curve at the decision threshold for the statistical eye to the left

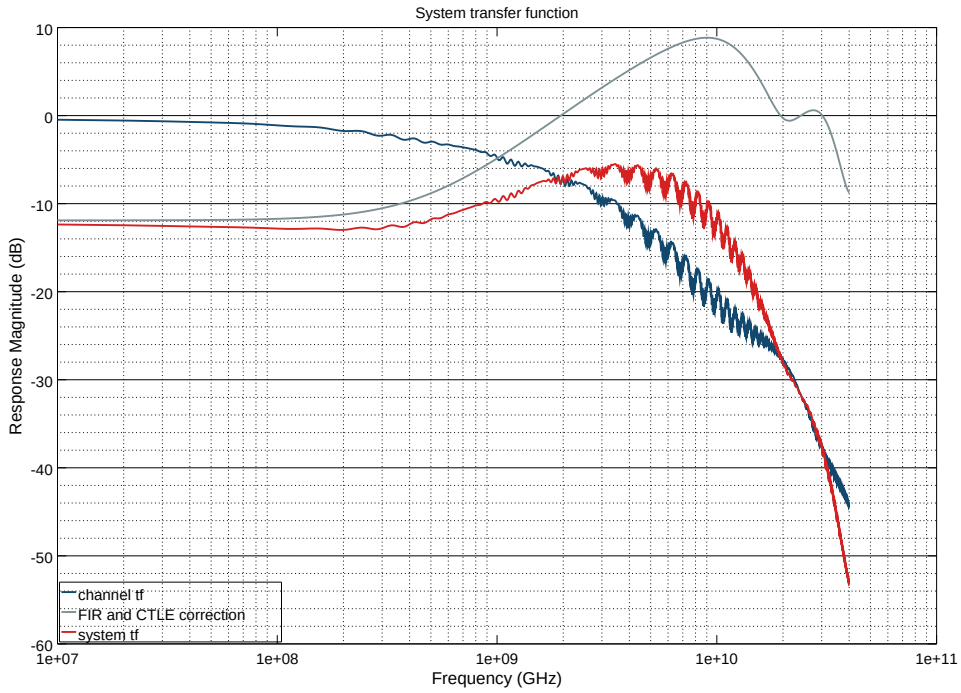

Figure 6.17: System transfer function for the chosen equalization settings (without DFE)

6.4 Influence of the sampler ISF

The major aim of developing the framework presented here is twofold. First, it shall be used to verify the functional correctness and specification conformity of serializer systems. The previous section highlighted some aspects of this process with respect to equalization provisioning and choices. The second goal, however, is to also allow for fast design space exploration as a tool for future serializer designs. As an example, the influence of the receiver sampler aperture on overall system performance will exemplify this aspect of the framework.

The same simulation as carried out in setting #1 is repeated here, only this time the aperture is decreased from the 4 ps of the previous version to 8 ps. This degradation of sampler aperture might for instance result from an IR drop on the supply or, as shown in section 4.5, due to lowered rise and fall times of the clock signal as they typically result from additional capacitive load on the clock tree (i.e. as a result of the post layout extraction) or from insufficient capacitive decoupling of the CMOS clock buffers.

Table 6.2 summarizes the results from the previous and this new simulation run.

Aperture	$V_{\mathrm{lo/hi}}$	$j_{\mathrm{d,pk-pk}}$	K_{PD}	f_{cdr}	$\sigma_{\mathrm{j,cdr}}$	e_{w}	e_{h}
4 ps	±87.1 mV	10 ps	9.15	10.1 MHz	0.216 ps	57.3 ps	116.9 mV
8 ps	±87.2 mV	11 ps	8	8.8 MHz	0.231 ps	34.2 ps	80.78 mV

Table 6.2: OpenMGT/OCM Analysis results for 10 Gbps operation of the presented serializer architecture

The equalization properties should not and are not affected by the change in aperture width as long as the correct bits are still resolved - a property checked by the transient simulation run itself outside the domain of numeric post processing. However, since the CDR phase detector gain directly depends on the sampler ISF, it consequently decreases to a lower value and with it the CDR bandwidth. This in turn increases the random jitter. The voltage noise on the other hand is of course not affected unless the physically reasonable reduction of sampler self noise due to the smaller sampling bandwidth would also be taken into account (which has not been done here). Compared to these mild changes in subcomponent metrics, however, the final statistical, post aperture eye shows a much more severe degradation, both in vertical and horizontal dimension. The strong decrease in eye width is a consequence of the weighting process of the now broadened aperture. The pre-aperture eye can be conceived as the eye as evaluated by a Dirac like sampler while the post-aperture eye, by convolution, factors in the nonideal voltage resolution process of the sampler. The same argument can be made for the decrease in eye height. The post aperture eye height needs to take into account the contributions of the input voltage at various time instants. The post aperture eye thus represents the ISF weighted average of the pre-aperture voltages seen over the ISF duration. The resulting iso-BER plots of the

two simulation runs can be seen in figures 6.18 and 6.19. Note that unlike the PDF of a Gaussian process with an equally detrimental effect to the eye, the increase in ISF does not place the iso lines closer together (or equivalently put: decrease the slope of the bathtub curve towards lower BER) but instead widens the low BER region as also caused by any other deterministic effect.

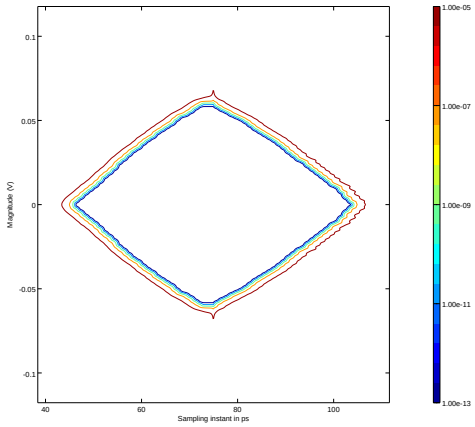

Figure 6.18: Iso BER lines resulting from the statistical eye with simulation settings #1 and an aperture of 4 ps

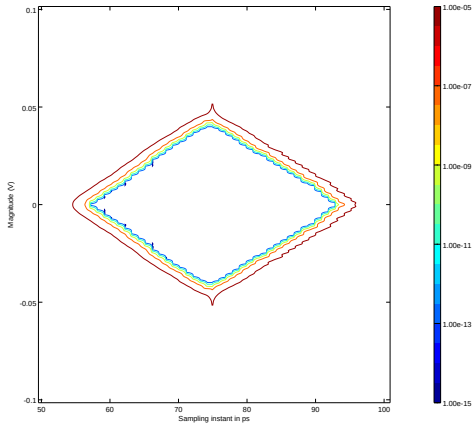

Figure 6.19: Iso BER lines resulting from the statistical eye with simulation settings #1 and an aperture of 8 ps

This clearly highlights the importance of a well-controlled and checked sampler aperture and the importance of the sampler model as presented in section 4.5.

6.5 Scaling channel geometry with the Johnson Signal Model

With the same simulation presets as given in the 10 Gigabit analysis from above, table 6.3 summarizes the results as obtained from the various simulation runs for different bisectional areas of the Johnson signal model of a coaxial cable. Table 6.4 summarizes the results as obtained from the various simulation runs for different cable lengths of the coaxial cable model with the area constrained to 0.35 mm^2. In both cases, the equalization efforts of setting #1 from the 10G analysis are carried out and the post processing results for eye width and height gathered.

Especially scaling with area follows the functional form of equation 3.9 in section 3.2 quite nicely (with the fit parameters being $B_0 = 1.25$ and $\kappa = 0.63$) - at least if the combined figure of merit $M = e_w \cdot e_h$ or only eye width is compared against it. The scaling trend with L for the few points observed looks quite more linear than the prediction with the characteristic strong decrease in slope towards higher values of L. For small channel lengths, a saturation can be observed which will mainly be due to the limited swings within the receiver itself rather than being a consequence of the physical channel properties.

Area	−30 dB **Frequency**	e_w	e_h	M (mV · ps)
0.05 mm^2	8.1 GHz	28.5 mV	29.3 ps	0.83 k
0.077 mm^2	9.8 GHz	44 mV	42.5 ps	1.87 k
0.12 mm^2	11.7 GHz	68 mV	49.2 ps	3.34 k
0.17 mm^2	13 GHz	87 mV	49.2 ps	4.28 k
0.35 mm^2	16 GHz	117 mV	53.5 ps	6.25 k
1.4 mm^2	20 GHz	147 mV	56.1 ps	8.24 k
4.4 mm^2	22.3 GHz	194 mV	55.1 ps	10.68 k
9 mm^2	23.2 GHz	200 mV	60.5 ps	12.1 k

Table 6.3: OpenMGT/OCM analysis of the Johnson Signal Model coaxial cable for different bisectional areas A and the equalization approach of setting #1

Area	−30 dB **cutoff**	e_w	e_h	M (mV · ps)
1.5 m	9.71 GHz	31 mV	26.8 ps	0.83 k
1.25 m	12.2 GHz	56 mV	42.7 ps	2.39 k
1 m	16.1 GHz	116 mV	53.5 ps	6.2 k
0.75 m	22.8 GHz	173 mV	60.4 ps	10.4 k
0.5 m	36.9 GHz	191 mV	63 ps	12.03 k

Table 6.4: OpenMGT/OCM analysis of the Johnson Signal Model coaxial cable for different cable lengths L and the equalization approach of setting #1

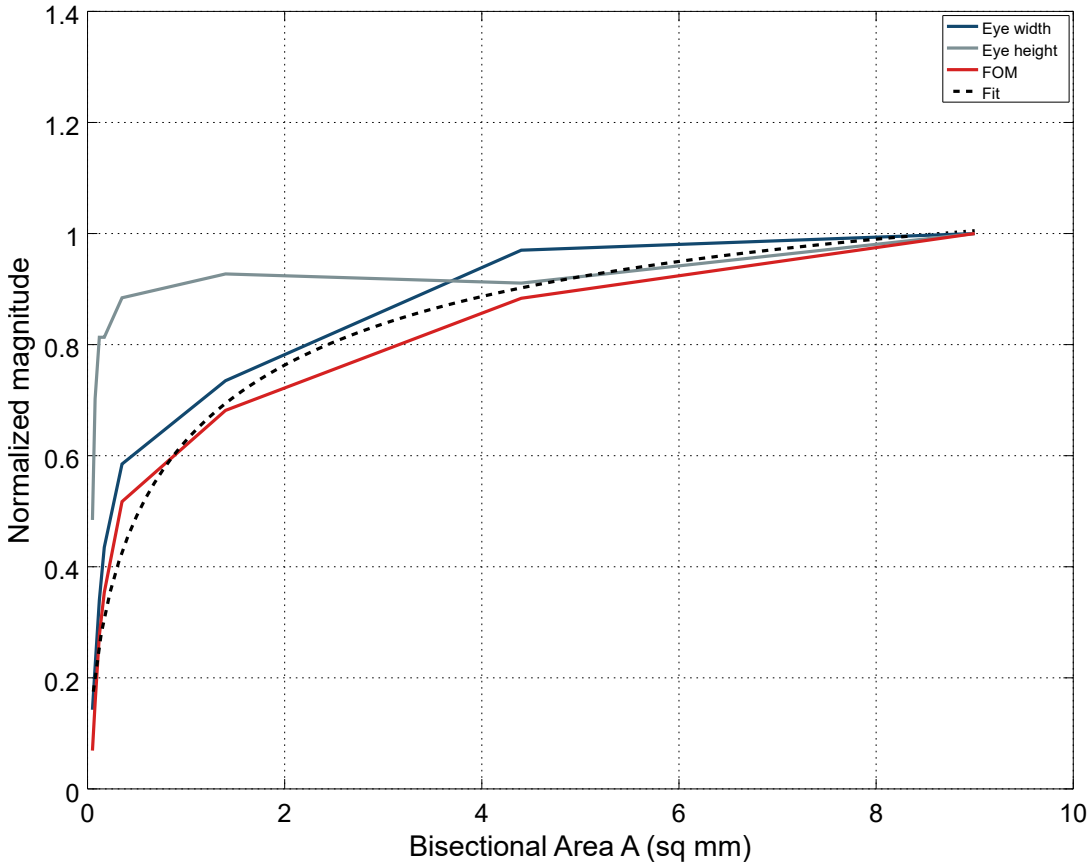

Figure 6.20: Normalized eye height, width and figure of merit M versus bisectional area A of the Johnson signal model coaxial cable

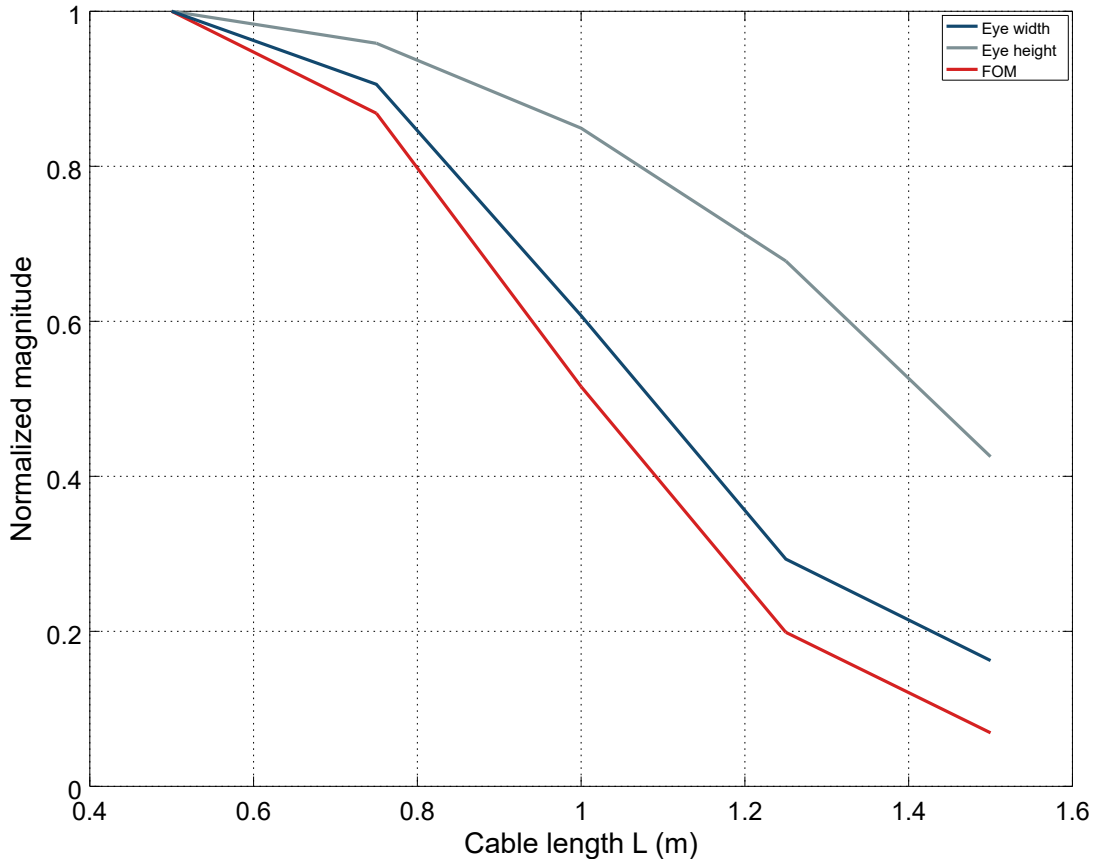

Figure 6.21: Normalized eye height, width and figure of merit M versus length L of the Johnson signal model coaxial cable

The data rate of the serializer here is chosen to be fixed. A more detailed procedure could also increase the rate of transmission with some sort of search algorithm and would basically use the same kind of information - eye height and width - to converge to this final, arguably lower data rate level. The major item of interest here, however, are the scaling trends with geometry rather than the actual bit rate capacity magnitudes. This also permits the analysis presented here which is carried out more easily and much faster.

In summary, the simulations indicate that the assumption of a globally acceptable, maximum level of attenuation as used in chapter 3 for predicting the underlying scaling laws of transmission channel geometry with respect to the achievable bit rate is acceptable. Since equalization to the largest part attenuates the signal frequencies up to a certain factor above the Nyquist frequency to a common level, this outcome seems very reasonable.

7 Conclusion and Outlook

This work presented a design and analysis framework for modelling and implementing a multigigabit serializer design. It is based on previously published analysis and modelling ideas for the various serializer subcomponents but realizes and combines them in a new and unique way to be compatible with the context of real number model based simulation and verification. The framework strictly separates deterministic and random effects in order to minimize the negative impact of statistical processes on analysis time. Additionally, the ansatz implicitly ensures consistency between different modelling views and the actual implementation of a subcomponent, especially between real number and full-analog realms. Due to the resulting, superior simulation performance, the framework thus allows to analyze a broad range of serializer system properties that were previously intractable by a single design and simulation environment.

The strong entanglement of the serializer with higher OSI protocol layers underlined the requirement of having meaningful, accurate, yet very fast simulation models at disposal to allow for higher level integration in larger systems and to enable the development of complex link protocols with automatic equalization support. To this end, the SystemVerilog real number modelling approach in conjunction with VerilogAMS mixed signal simulation was used to conceive the openMGT framework in collaboration with another dissertation [38]. Its main ideas are a top-down, digital first implementation approach together with a flexible model view on each system subcomponent. The modelling views can be either of purely functional (VerilogHDL), real number signal flow or electrical (SPICE/SPECTRE) nature depending on modelling scope and required accuracy. This mechnism is used to keep modelling and implementation consistent. It also introduces the concept of interoperable testbenches for real number and all analog representation which may serve as both a means of analog component verification and real number model extraction. On the basis of well-established theories, the process was exemplified with the computationally most challenging parts of a serializer system: the transmission channel and the high speed receiver samplers.

By extending the commercial simulation tools with optimized and parallelized custom C code, the compuational flexibility of *Octave*, the SystemVerilog DPI extension and by optimizing the SystemVerilog modelling process, the resulting real number singal flow models simulate approximately a factor of ***a thousand*** times faster than their all analog counterparts while no loss in accuracy with respect to the extracted subcomponent

parameters is observed. The modellig infrastructure also allows to use parameterizable, numeric models in order to use the resulting simulation framework for a more general analysis of the so-called *transmission channel bit rate capacity*.

Moreover, the Octave extension to the SystemVerilog openMGT framework via DPI, dubbed *OCM*, also enables the powerful numeric postprocessing required for serializer budgeting and design analysis. It provides mathematical algorithms not customarily found within SystemVerilog itself and ways to parallelize vector and matrix based numeric computation to drastically decrease post processing runtime. OCM therefore not only allows the flexible definition of models, but also advanced data acquisition and post processing functionality along with a mechanism to keep the model definitions used in transient simulations fully consistent with the data base of post processing analysis.

One particular application to OCM presented here are physical transmission channels and their bit rate capacity. Channels are very difficult to model, involving many freely selectable parameters and geometries. This task is therefore often delegated to advanced, commerical microwave engineering software tools. For more comparable analyses in the context of the bit rate capacity as use in HPC technology projections, the Johnson signal model of a coaxial line is presented here as an alternative and implemented in OCM. It allows to capture more physical channel effects than previously and is thus used as a basis for a new projection attempt. Even without considering the serializer system itself, general trends of the bit rate capacity in the multigigabit regime with respect to channel impedance, bisectional area and length are shown. These results ammend previous work done in this context. The trends derived indicate a scaling with bisectional area that is less favorable than previously anticipated. It also highlights the optimality of signalling in the impedance regime around 45 to 60 Ω. On the other hand, it can also be seen that the bit rate capacity may actually downscale less drastically with length than the previously postulated, geometric trend.

The design and modelling framework is supplemented by a serializer link budgeting algorithm. It combines the insights of the foregoing chapters and captures all deterministic effects of the system such as duty cycle distortion, residual offsets or imperfect equalization in a transient simulation run. A strict separation of deterministic effects from random timing noise and random voltage noise enables fast design space exploration and analysis. The OCM extension allows model data consistency by working within a single tool environment only.

The implementation presented in this work is based on the well-known peak distortion algorithm. It is a statistical analysis to reduce the required transient simulation time by a factor of roughly one million to achieve the same level of statistical eye diagram accuracy. The algorithm is implemented in Octave and uses its above mentioned parallelization capabilities to speed up this process.

The budgeting procedure includes effects of timing jitter, voltage noise and a linearized

phase state model for the clock recovery circuit to arrive at an estimate for the statistical eye diagram as it ought to be observed by the receiver. It does so by considering timing jitter entirely in frequency domain (as opposed to time domain descriptions in traditional approaches). It also describes the CDR with its phase state model based on well established theory enriched by information otherwise not available to modelling attempts outside of OCM. This is especially true with respect to sampler performance metrics such as the impulse sensitivity function or to deterministic jitter resulting from residual intersymbol interference.

The framework is then used to assess the performance of a serializer design for 10 Gigabit Ethernet as it was codeveloped in a team and along with this thesis. Different options of equalization and their effect on overall eye opening are analyzed and the presented serializer system is shown to be compliant with the 10G Ethernet channel model.

In order to highlight the frameworks capability of supporting design space exploration efforts and the importance of sampler aperture modelling, the impact of the sampler impulse sensitivity function on overall system performance is demonstrated.

Also, the Johnson signal model is used in conjunction with this system to gain more realistic insight into the scaling properties of bit rate capacity with bisectional area and length. The heuristic predictions resulting from the postulation of a globally acceptable level of attenuation leads to very similar scaling trends when compared with the actual serializer system which covers a much greater variety of contributions to the final outcome.

In order to verify the budgeting approach presented here, laboratory measurements of the serializer system have to be carried out and comapred to the numeric results once the chip returns from fabrication. In this context, the jitter projections made by the frequency space approach are, of course, of special interest. Also the framework may and shall be extended by power aware models to facilitate the conception of power saving states and higher level protocol support for more energy efficient network protocols. It also awaits integration into higher level, network protocol design efforts and must proof its value in finding new concepts for equalization preset negotiation, dynamic link frequency or width scaling or the implementation of advanced power down and idle states to increase I/O power efficiency in future designs.

List of Figures

List of Tables

Appendix A - System component metric overview

Symbol	Description	Source
Overall system		
BER	Target bit error rate	Set by serializer application requirements, especially error correction capabilities
e_{w}	Eye width at target BER	Result of analysis
e_{h}	Eye height at target BER	Result of analysis
UI	Unit interval - time duration of a single symbol as modulated onto the transmission line	From data rate and line coding, application dependent
T_{env}	Environmental system temperature at which channel and termination noise is calculated	Constrained by budgeting
df_{ϕ}	Phase noise transfer function and power spectral density resolution used during post processing	Constrained by accuracy and performance requirementof the budgeting run
Power distribution networks		
$\sigma_{\mathrm{vn,pdn}}$	Power distribution network in-band RMS voltage noise	Specified for simple model or derived from $S_{\mathrm{VV}}(\omega)$
$\omega_{\mathrm{3db,pdn}}$	Power distribution network voltage noise bandwidth	Specified for simple model or derived from $S_{\mathrm{VV}}(\omega)$
$S_{\mathrm{I}}(\omega)$	Spectrally resolved current consumption of system attached to given PDN	From transient (RNM) simulation
$Z_{\mathrm{PDN}}(\omega)$	Spectrally resolved impedance of the complete PDN	From board and package level EM extraction and vendor supplied model data (capacitors, VRM)
Phase locked loops		

$S_{\phi\phi}(\omega)$	Phase noise power spectral density of PLL clock output including PDN effects	Measured by PSS noise analysis or derived from simple model
$\sigma_{j,ref/xco}$	Total RMS output jitter of the reference/ PLL XC oscillator measured within the frequency band of 0 to $f_{\mathrm{j,BW}}$	As given by datasheets or constrained by budgeting
$f_{\mathrm{j,BW,ref/xco}}$	Bandwidth across which $\sigma_{j,ref/xco}$ is accumulated by the reference / XC oscillator	As given by datasheets or constrained by budgeting
$f_{\mathrm{osc,ref/xco}}$	Center oscillation frequency of reference / XC oscillator	As given by datasheets or constrained by budgeting
$f_{\mathrm{osc},\gamma,\mathrm{ref/xco}}$	Spectral width of center oscillation frequency of reference / XC oscillator if Lorentzian phase noise model is used	As given by datasheets or constrained by budgeting
K_{PD}	Phase detector gain	Constrained by budgeting or derived from testbench
K_{LP}	Filter stage gain	Constrained by budgeting or derived from testbench
K_{XCO}	Voltage/Digital controlled oscillator gain	Constrained by budgeting or derived from testbench
K_{D}	Divider feedback gain	Given by the ratio of $f_{\mathrm{osc,xco}}$ and $f_{\mathrm{osc,ref}}$
τ_{LP}	Low pass filter time constant	Constrained by budgeting
Transmitter		
Z_{TX}	Transmitter output impedance	Usually a tuning range, constrained by specification and tuning requirements
$V_{\mathrm{tx,pp}}$	Output signal swing	Constrained by output buffer architecture and technology node
$t_{\mathrm{r/f}}$	Output signal rise and fall times as measured directly at the transmitter output	Some standards define/constrain this parameter, can be derived from budgeting
$H_{\mathrm{vdd2o}}(\omega)$	Worst case power supply rejection ratio mask of the transmitter output buffer stage	Constrained by budgeting

$S_{\mathrm{VV,int}}(\omega)$	Output voltage noise power spectral density from TX internal noise sources	Either a mask for budgeting or extracted from schematic based simulations
$S_{\phi\phi,\mathrm{tx,clk}}(\omega)$	Phase noise power spectral density added by TX clock distribution noise sources	Either a mask for budgeting or extracted from schematic based simulations
$H_{\mathrm{vdd2,tx}\phi}(\omega)$	TX clock distribution power supply sensitivity	Either a mask for budgeting or extracted from schematic based simulations
Receiver		
Z_{RX}	Receiver input impedance	Usually a tuning range, constrained by specification and tuning requirements
$H_{\mathrm{vdd2o}}(\omega)$	Worst case power supply rejection ratio mask of the receiver amplifier chain	Constrained by budgeting
$S_{\mathrm{VV,int}}(\omega)$	Receiver voltage noise power spectral density from RX internal noise sources	Either a mask for budgeting or extracted from schematic based simulations
$S_{\phi\phi,\mathrm{rx,clk}}(\omega)$	Phase noise power spectral density added by RX clock distribution noise sources	Either a mask for budgeting or extracted from schematic based simulations
$H_{\mathrm{vdd2,rx}\phi}(\omega)$	RX clock distribution power supply sensitivity	Either a mask for budgeting or extracted from schematic based simulations
Clock data recovery		
K_{PD}	Phase detector gain	Derived
N_{dly}	Total control loop delay in bit unit intervals	Fixed by architecture
K_{D}	Decimation gain	Fixed by decimation approach (i.e. box car or majority vote)
K_{DPC}	Digital to phase (phase interpolator) conversion gain	Actually an attenuation and given by the phase interpolator resolution
K_{P}	CDR proportional gain	Usually an adjustable quantity, found in-system for best performance

K_{I}	CDR integral gain	Usually an adjustable quantity, found in-system for best performance
Peak Distortion Algorithm		
$vr_{\mathrm{pda,thr}}$	Relative magnitude of an SBR tap compared to its maximum value to still be considered for PDA analysis	Fixed by accuracy requirements
$vr_{\mathrm{pda_v res}}$	PDA analysis SBR voltage resolution	Fixed by accuracy requirements

Table .1: OCM link budgeting variables as defined and used throughout this text

Appendix B - Design budgeting constraints

Symbol	Value	Notes
Overall system		
BER	10^{-13}	Slight overconstraint to the usual 10^{-12} for margin
UI	In variation	
T_{env}	300° K	Room temperature
df_{ϕ}	100 kHz	
PLL Power distribution network		
$\sigma_{\text{vn,pdn}}$	5 mV	
$\omega_{\text{3db,pdn}}$	2 GHz	High frequency cutoff due to on-die capacitance of low ESR
TX/RX Power distribution network		
$\sigma_{\text{vn,pdn}}$	5 mV	
$\omega_{\text{3db,pdn}}$	1 GHz	
Low jitter phase locked loop model		
$\sigma_{j,ref}$	750 fs	
$f_{\text{j,BW,ref}}$	1 MHz	
$f_{\text{osc,ref}}$	100 MHz	
$f_{\text{osc},\gamma,\text{ref}}$	250 kHz	
$\sigma_{j,ref}$	500 fs	
$f_{\text{j,BW,ref}}$	10 MHz	
$f_{\text{osc,ref}}$	5 GHz	
$f_{\text{osc},\gamma,\text{ref}}$	250 kHz	
K_{PD}	500	
K_{LP}	1	
K_{XCO}	10 GHz/V	
K_{D}	50	
τ_{LP}	20 ns	

Transmitter		
Z_{TX}	50 Ω single ended, 100 Ω differential	
$V_{tx,pp}$	800 mV differential with transmitter terminated with matched load	
$t_{r/f}$	20 ps pre ESD compensation	
$H_{vdd2o}(\omega)$	Simple first order high pass model with onset frequency of 500 MHz minimal susceptibiliy of 10^{-3} and maximum of 10^{-2}	
$S_{VV,int}(\omega)$	Simple low pass noise model with 1.5 mV_{rms} within a band of 17 GHz	
$S_{\phi\phi,tx,clk}(\omega)$	as reported by PLL model	
$H_{vdd2,tx\phi}(\omega)$	-	unused
Receiver		
Z_{RX}	50 Ω single ended, 100 Ω differential	
$H_{vdd2o}(\omega)$	Simple first order high pass model with onset frequency of 500 MHz minimal susceptibiliy of 10^{-4} and maximum of 10^{-2}	CML amplifier stages with better noise rejection than SST segments of transmitter at low frequencies
$S_{VV,int}(\omega)$	Simple low pass noise model with 2 mV_{rms} within a band of 17 GHz	
$S_{\phi\phi,rx,clk}(\omega)$	As reported by PLL	
$H_{vdd2,rx\phi}(\omega)$	-	unused
Clock data recovery		
N_{dly}	32	Given by architecture, CDR running in clk8 domain
K_D	8	Simple majority voting
K_{DPC}	$\frac{1}{2^6}$	Given by PI resolution
K_P	$\frac{1}{2^8}$	not optimized in simulation
K_I	$\frac{1}{2^{19}}$	not optimized in simulation
Peak Distortion Algorithm		
$vr_{pda,thr}$	not used but fixed to 4 precursors and 20 postcursors	

$vr_{\mathrm{pda_v res}}$	2.5 mV	

Table .2: OCM link budgeting variable values for budgeting runs as presented in chapter 6

Bibliography

[1] Accellera. 'SystemVerilog 3.1a Language Reference Manual'. In: (2004), p. 586. URL: http://www.eda.org/sv/SystemVerilog%7B%5C_%7D3.1a.pdf.

[2] Accellera. *Verilog-AMS Language Reference Manual v2.4*. 2014. URL: http://www.accellera.org/downloads/standards/v-ams.

[3] David E. Bockelman and William R. Eisenstadt. 'Combined differential and common-mode scattering parameters: theory and simulation'. In: *IEEE Transactions on Microwave Theory and Techniques* 43.7 pt 1 (1995), pp. 1530–1539. ISSN: 00189480. DOI: 10.1109/22.392911. arXiv: 95 [0018-9480].

[4] Bulzacchelli. 'A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology'. In: *IEEE Journal of Solid-State Circuits* 41.12 (Dec. 2006), pp. 2885–2900. ISSN: 0018-9200. DOI: 10.1109/JSSC.2006.884342. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4014602.

[5] Anthony Chan Carusone. 'An equalizer adaptation algorithm to reduce jitter in binary receivers'. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 53.9 (2006), pp. 807–811. ISSN: 10577130. DOI: 10.1109/TCSII.2006.881161.

[6] B.K. Casper, M. Haycock, and R. Mooney. 'An accurate and efficient analysis method for multi-Gb/s chip-to-chip signaling schemes'. In: *2002 Symposium on VLSI Circuits. Digest of Technical Papers (Cat. No.02CH37302)* (2002), pp. 54–57. DOI: 10.1109/VLSIC.2002.1015043.

[7] Santanu Chaudhuri et al. 'Jitter Amplification Characterization of Passive Clock Channels at 6.4 and 9.6 Gb/s'. In: *IEEE Electrical Performance of Electronic Packaging* (2006), pp. 35–38.

[8] J Chen. *Mixed-Signal Methodology Guide*. 2014, p. 408. ISBN: 9781300035206.

[9] James Chen. 'Self-calibrating on-chip interconnects [Dissertation]'. PhD thesis. Stanford University, 2012.

[10] Norbert Fliege and Markus Gaida. *Signale und Systeme: Grundlagen und Anwendungen mit MATLAB*. 1st ed. Schlembach, 2008, p. 355. ISBN: 978-3935340427.

[11] Gautam R. Gangasani et al. ‘A 16-Gb/s Backplane Transceiver With 12-Tap Current Integrating DFE and Dynamic Adaptation of Voltage Offset and Timing Drifts in 45-nm SOI CMOS Technology’. In: *IEEE Journal of Solid-State Circuits* 47.8 (Aug. 2012), pp. 1828–1841. ISSN: 0018-9200. DOI: 10.1109/JSSC.2012.2196313. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6244847%20http://ieeexplore.ieee.org/document/6244847/.

[12] GNU. *Octave Scientific Programming Language*. 2017. URL: https://www.gnu.org/software/octave/.

[13] Karthik Gopalakrishnan et al. ‘3.4 A 40/50/100Gb/s PAM-4 Ethernet transceiver in 28nm CMOS’. In: *Digest of Technical Papers - IEEE International Solid-State Circuits Conference* 59 (2016), pp. 62–63. ISSN: 01936530. DOI: 10.1109/ISSCC.2016.7417907.

[14] PCI-SIG Group. *PCI Express Base Specification Revision 3.1a*. 2015.

[15] Amr Amin Hafez, Ming-Shuan Chen, and Chih-Kong Ken Yang. ‘A 32-to-48Gb/s serializing transmitter using multiphase sampling in 65nm CMOS’. In: IEEE, 2013, pp. 38–39. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6487627.

[16] H. Hatamkhani et al. ‘Power-centric design of high-speed I/Os’. In: *2006 43rd ACM/IEEE Design Automation Conference* 1 (2006), pp. 867–872. ISSN: 0738-100X. DOI: 10.1109/DAC.2006.229252. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1688919.

[17] IEEE. *IEEE Standard for Ethernet, IEEE Standard 802.3-2015*. 2015.

[18] IEEE. *IEEE Standard for SystemVerilog – Unified Hardware Design, Specification and Verification Language, IEEE Standard 1800-2012*. 2012.

[19] IEEE Computer Society. *IEEE Standard Verilog Hardware Description Language*. 2001, p. 791.

[20] Intel. *Intel Xeon Phi Processors*. 2017. URL: http://www.intel.com/content/www/us/en/products/processors/xeon-phi/xeon-phi-processors.html (visited on 02/28/2017).

[21] ITRS. *International Technology Roadmap Service Report 2013*. Tech. rep. 2013. URL: http://www.itrs2.net/.

[22] M. Jeeradit et al. ‘Characterizing sampling aperture of clocked comparators’. In: *IEEE Symposium on VLSI Circuits, Digest of Technical Papers* (2008), pp. 64–65. ISSN: 1424418046. DOI: 10.1109/VLSIC.2008.4585955.

[23] Jhih-Yu Jiang et al. ‘100Gb/s ethernet chipsets in 65nm CMOS technology’. In: IEEE, 2013, pp. 120–121. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6487663.

[24] Howard Johnson and Martin Graham. *High-Speed Signal Propagation*. 1st ed. Upper Saddle River, NJ 07458: Prentice Hall, 2003. ISBN: 0-13-084408-X.

[25] Kambiz Kaviani et al. 'A 0.4-mW/Gb/s near-ground receiver front-end with replica transconductance termination calibration for a 16-Gb/s source-series terminated transceiver'. In: *IEEE Journal of Solid-State Circuits* 48.3 (2013), pp. 636–648. ISSN: 00189200. DOI: 10.1109/JSSC.2013.2242714.

[26] Kambiz Kaviani et al. 'A Tri-modal 20-Gbps/link differential/DDR3/GDDR5 memory interface'. In: *IEEE Journal of Solid-State Circuits*. Vol. 47. 4. 2012, pp. 926–937. ISBN: 978-4-86348-166-4. DOI: 10.1109/JSSC.2012.2185370.

[27] Jaeha Kim, Brian S. Leibowitz, and Metha Jeeradit. 'Impulse sensitivity function analysis of periodic circuits'. In: *2008 IEEE/ACM International Conference on Computer-Aided Design* (2008), pp. 386–391. DOI: 10.1109/ICCAD.2008.4681602. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4681602.

[28] Jihwan Kim et al. 'A 16-to-40Gb/s quarter-rate NRZ/PAM4 dual-mode transmitter in 14nm CMOS'. In: *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*. Vol. 58. IEEE, 2015, pp. 60–61. ISBN: 9781479962235. DOI: 10.1109/ISSCC.2015.7062925. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=7062925.

[29] Peter Kogge et al. 'ExaScale Computing Study : Technology Challenges in Achieving Exascale Systems'. In: *Government PROcurement* TR-2008-13 (2008), p. 278. DOI: 10.1.1.165.6676.

[30] Stefan Kosnac. 'Design-Aspects of a Decision Feedback Equalizer in a 28nm Technology [Master Thesis]'. PhD thesis. University of Heidelberg, 2016, p. 68.

[31] Kenneth S. Kundert. 'Verification of Bit-Error Rate in Bang-Bang Clock and Data Recovery Circuits'. In: (2010), pp. 1–22.

[32] Jri Lee, Ming-shuan Chen, and Huai-de Wang. 'Design and Comparison of Three 20-Gb/s Backplane Transceivers for Duobinary, PAM4 and NRZ Data'. In: 43.9 (2008), pp. 2120–2133.

[33] Chris Madden. 'Jitter modeling in statistical link simulation'. In: *2008 IEEE International Symposium on Electromagnetic Compatibility* 4 (2008), pp. 1–4. DOI: 10.1109/ISEMC.2008.4652155. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4652155.

[34] M. Mansuri and Chih-Kong Ken. 'Jitter optimization based on phase-locked loop design parameters'. en. In: *IEEE Journal of Solid-State Circuits* 37.11 (Nov. 2002), pp. 1375–1382. ISSN: 0018-9200. DOI: 10.1109/JSSC.2002.803935. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1046079.

[35] Christian Menolfi et al. ‘A 14Gb/s high-swing thin-oxide device SST TX in 45nm CMOS SOI’. In: IEEE, 2011, pp. 156–158. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=5746262.

[36] D.A.B. Miller. ‘Limit to the bit-rate capacity of electrical interconnects from the aspect ratio of the system architecture’. In: *Journal of parallel and distributed computing* 41.1 (1997), pp. 42–52. URL: http://www.sciencedirect.com/science/article/pii/S074373159691285X.

[37] J. Montanaro et al. ‘A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor’. In: *IEEE Journal of Solid-State Circuits* 31.11 (1996), pp. 1703–1714. ISSN: 0018-9200. DOI: 10.1109/JSSC.1996.542315. URL: http://ieeexplore.ieee.org/ielx4/4/11741/00542315.pdf?tp=%7B%5C&%7Darnumber=542315%7B%5C&%7Disnumber=11741.

[38] Markus Müller. ‘System on chip interconnects (working title) [disertation]’. PhD thesis. 2017.

[39] Takanori Nakao et al. ‘An equalizer-adaptation logic for a 25-Gb/s wireline receiver in 28-nm CMOS’. In: *Proceedings of the 2013 IEEE Asian Solid-State Circuits Conference, A-SSCC 2013* (2013), pp. 217–220. DOI: 10.1109/ASSCC.2013.6691021.

[40] Meisam Honarvar Nazari and Azita Emami-Neyestanak. ‘A 15-Gb/s 0.5-mW/Gbps two-tap DFE receiver with far-end crosstalk cancellation’. In: *IEEE Journal of Solid-State Circuits* 47.10 (2012), pp. 2420–2432. ISSN: 00189200. DOI: 10.1109/JSSC.2012.2203870.

[41] Ethiopia Nigussie et al. ‘Semi-Serial On-Chip Link Implementation for Energy Efficiency and High Throughput’. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 20.12 (Dec. 2012), pp. 2265–2277. ISSN: 1063-8210, 1557-9999. DOI: 10.1109/TVLSI.2011.2170228. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6062666.

[42] ‘NRZ Bandwidth - HF Cutoff vs. SNR’. In: *Note, Maxim Integrated Application* 04.2 (2008).

[43] Nvidia. *Nvidia Tesla GPU*. 2017. URL: http://www.nvidia.com/object/tesla-servers.html (visited on 02/28/2017).

[44] Dan Oh, Jihong Ren, and Sam Chang. ‘Hybrid statistical link simulation technique’. In: *IEEE Transactions on Components, Packaging and Manufacturing Technology* 1.5 (2011), pp. 772–783. ISSN: 21563950. DOI: 10.1109/TCPMT.2011.2118209.

[45] Dan Oh et al. ‘Prediction of system performance based on component jitter and noise budgets’. In: *IEEE Topical Meeting on Electrical Performance of Electronic Packaging* (2007), pp. 33–36. DOI: 10.1109/EPEP.2007.4387116.

[46] OIF. *Common Electrical I/O (CEI) - Electrical and Jitter Interoperability agreements for 6G+ bps, 11G+ bps and 25G+ bps I/O*. 2014, p. 300.

[47] PCI-SIG. ‘PCI Express Architecture Jitter Modeling 1.0RD’. 2004.

[48] Ping-Hsuan Hsieh et al. 'A 28-Gb/s 4-Tap FFE/15-Tap DFE Serial Link Transceiver in 32-nm SOI CMOS Technology'. In: *IEEE Journal of Solid-State Circuits* 47.12 (Dec. 2012), pp. 3232–3248. ISSN: 0018-9200, 1558-173X. DOI: 10.1109/JSSC.2012.2216414. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6327618.

[49] Rick Poore. 'Overview on Phase Noise and Jitter'. In: *Agilent EEsof EDA* (2001).

[50] John W. Poulton et al. 'A 0.54 pJ/b 20Gb/s ground-referenced single-ended short-haul serial link in 28nm CMOS for advanced packaging applications'. In: IEEE, 2013, pp. 404–405. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=6487789.

[51] Pozar. *Microwave engineering*. Fourth. John Wiley & Sons, 1998. ISBN: 0-471-17096-8.

[52] J.G. Proakis. *Digital Communications*. Third Edit. McGraw-Hill, 1995.

[53] Fangyi Rao and Sammy Hindi. 'Frequency domain analysis of jitter amplification in clock channels'. In: *2012 IEEE 21st Conference on Electrical Performance of Electronic Packaging and Systems, EPEPS 2012* 2.5 (2012), pp. 51–54. DOI: 10.1109/EPEPS.2012.6457841.

[54] Behzad Razavi. 'The StrongARM latch [A Circuit for All Seasons]'. In: *IEEE Solid-State Circuits Magazine* 7.2 (2015), pp. 12–17. ISSN: 19430582. DOI: 10.1109/MSSC.2015.2418155.

[55] Justin Redd and Craig Lyon. 'Spectral content of NRZ test patterns'. In: *Maxim Integrated Application Note* 49.18 (2004), pp. 67–72. ISSN: 00127515.

[56] Sven Schatral. 'Yet unknown [Dissertation]'. PhD thesis. 2017.

[57] S. Shekhar et al. 'Design considerations for low-power receiver front-end in high-speed data links'. In: *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference* (2013), pp. 1–8. ISSN: 08865930. DOI: 10.1109/CICC.2013.6658406. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6658406.

[58] Jeff Sonntag and John Stonick. 'A digital clock and data recovery architecture for multi-gigabit/s binary links'. In: *Proceedings of the Custom Integrated Circuits Conference* 2005.8 (2005), pp. 532–539. ISSN: 08865930. DOI: 10.1109/CICC.2005.1568725.

[59] Rick Stevens and Andrew White. 'Scientific Grand Challenges - Exascale Computing'. In: *Workshop on Architectures and Technology for Extreme Scale Computing* (2009). URL: http://science.energy.gov/ascr/news-and-resources/workshops-and-conferences/grand-challenges/.

[60] V. Stojanovic and M. Horowitz. 'Modeling and analysis of high-speed links'. In: IEEE, 2003, pp. 589–594. ISBN: 978-0-7803-7842-1. DOI: 10.1109/CICC.2003.1249467. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1249467.

[61] Vladimir Stojanovic, Amir Amirkhany, and Mark A. Horowitz. 'Optimal linear precoding with theoretical and practical data rates in high-speed serial-link backplane communication'. In: vol. 5. IEEE, 2004, pp. 2799–2806. URL: http://ieeexplore.ieee.org/xpls/abs%7B%5C_%7Dall.jsp?arnumber=1313040.

[62] T. Toifl et al. 'A 22-Gb/s PAM-4 Receiver in 90-nm CMOS SOI Technology'. en. In: *IEEE Journal of Solid-State Circuits* 41.4 (Apr. 2006), pp. 954–965. ISSN: 0018-9200. DOI: 10.1109/JSSC.2006.870898. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1610640.

[63] M Tsuk et al. 'An electrical-level superposed-edge approach to statistical serial link simulation'. In: *Computer-Aided Design - Digest of Technical Papers, 2009. ICCAD 2009. IEEE/ACM International Conference on* (2009), pp. 717–724. ISSN: 1092-3152. DOI: 10.1145/1687399.1687533.

[64] A X Widmer and P A Franaszek. 'A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code'. In: *IBM Journal of Research and Development* 27.5 (1983), pp. 440–451. ISSN: 0018-8646. DOI: 10.1147/rd.275.0440.

[65] Ricki Dee Williams and Theresa Sze. 'JEDEC Server Memory Roadmap'. In: 2012.

[66] Hyosup Won et al. 'An on-chip stochastic sigma-tracking eye-opening monitor for BER-optimal adaptive equalization'. In: *Proceedings of the Custom Integrated Circuits Conference* 2015-Novem (2015), pp. 15–18. ISSN: 08865930. DOI: 10.1109/CICC.2015.7338374.

[67] Arash Zargaran-yazd and Wendemagagnehu T Beyene. 'Discrete-Time Modeling and Simulation Considerations for High-Speed Serial Links'. In: (2014), pp. 165–168.