Design Techniques for High Performance Wireline Communication and Security Systems

by

Shiming Song

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering and Computer Science)
in The University of Michigan
2018

Doctoral Committee:

Associate Professor Zhengya Zhang, Chair
Assistant Professor Hessam Mahdavifar
Associate Professor Christopher Peikert
Professor S. Sandeep Pradhan

Shiming Song

shisong@umich.edu

ORCID iD: 0000-0001-6021-9061

To my family and friends

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Professor Zhengya Zhang, for all his guidance and advice throughout my study. I started my adventure into VLSI in an undergraduate level class taught by Zhengya, and since then I have learned from him about topics ranging from flip-flop timing to channel decoders. Later, I started my research in Zhengya's group and explored many advanced research areas. Zhengya has always been a great mentor. After years of working in the VLSI-SP group, I have become more hardworking, more curious, and always enthusiastic about the tech world.

I have also received help from many other members of the EECS department. I would like to thank Professor Michael Flynn and his group for valuable suggestions on circuit design and sharing with me testing equipments. Also, I would like to thank Professor Wei Lu for the opportunities to learn about advanced applications of memory circuits, and Professor Ehsan Afshari for the opportunities to learn about terahertz systems. I would like to thank my thesis committee, Professor Hessam Mahdavifar, Professor S. Sandeep Pradhan, and Professor Chris Peikert for all the suggestions and thoughtful questions along the way towards this thesis.

I am also grateful to my labmates and peer graduate students for all the inspiring discussions and valuable help. I would like to thank Thomas Chen for the helpful support on my research projects. I learned from him how to use a lot of the in-lab tools. I would like to thank Wei Tang for the great cooperation and many days of hard work together and constructive discussions in the lab. I would like to thank

Kyojin Choo for the great teamwork on our two-year project. I also learned from him substantial technique and experience on wireline transceivers. I am also grateful for the support from the whole VLSI-SP group. I'd like to thank Chester Liu, Sung-Gun Cho, Alex Lee, Teyuh Chou, Jie-Fang Zhang, Jacob Botimer and Reid Pinkham for always being supportive.

I would also like to thank my friends in Ann Arbor for exploring this beautiful town with me for the last 9 years since my freshman year.

Finally, I would like to thank my parents, for all the support, guidance and encouragement along the way.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ABSTRACT

As the amount of data traffic grows exponentially on the internet, towards thousands of exabytes by 2020, high performance and high efficiency communication and security solutions are constantly in high demand, calling for innovative solutions. Within server communication dominates todays network data transfer, outweighing between-server and server-to-user data transfer by an order of magnitude. Solutions for within-server communication tend to be very wideband, i.e. on the order of tens of gigahertz, equalizers are widely deployed to provide extended bandwidth at reasonable cost. However, using equalizers typically costs the available signal-to-noise ratio (SNR) at the receiver side. What is worse is that the SNR available at the channel becomes worse as data rate increases, making it harder to meet the tight constraint on error rate, delay, and power consumption. In this thesis, two equalization solutions that address optimal equalizer implementations are discussed. One is a low-power high-speed maximum likelihood sequence detection (MLSD) that achieves record energy efficiency, below 10 pico-Joule per bit. The other one is a phase-shaping equalizer design that suppresses inter-symbol interference at almost zero cost of SNR. The growing amount of communication use also challenges the design of security subsystems, and the emerging need for post-quantum security adds to the difficulties. Most of currently deployed cryptographic primitives rely on the hardness of discrete logarithms that could potentially be solved efficiently with a powerful enough quantum computer. Efficient post-quantum encryption solutions have become of substantial value. In this thesis a fast and efficient lattice encryption application-specific integrated circuit is presented that surpasses the energy efficiency of embedded processors by 4 orders of magnitude.

CHAPTER I

Introduction

The past decade has witnessed an explosive growth of new digital computing markets, largely thanks to the continuous breakthroughs and innovations in cloud-and-data-driven technologies, such as machine learning, and the internet of things, just to name a few. The mechanisms for transporting, storing, and securing data has thus sustained their roles as major commodities, along with the useful data themselves. Historically, the study of communication, storage and security has drawn wide attention, inspiring large volumes of both theoretical and experimental works, and even motivated a separate field of study named information theory *Cover and Thomas* (2012), the elegance of which has been of huge value on its own, owing to the talents and efforts of the many scientists and engineers since the seminal works by Shannon *Shannon* (2001).

Despite the many well-known masterworks and the unceasing effort of generations of talents, real world challenges in communication, data storage, and data security continue to rise and shift along the way, and closure is yet far from anywhere in the foreseeable future. In this thesis, attempts have been made to address some important aspects of communication and security, with no intention to make a thorough review or a universal solution.

1.1 High Speed Communication Overview

The growing demand for data-driven applications has been constantly pushing the underlying communication infrastructures and calling for almost an order of magnitude improvement in data rate in each generation of products.

Data Center IP Traffic, 2015–2020							
	2015	2016	2017	2018	2019	2020	CAGR 2015–2020
By Type (EB per Year)							
Data center to user	744	933	1,164	1,438	1,772	2,183	24.0%
Data center to data center	346	515	713	924	1,141	1,381	31.9%
Within data center	3,587	5,074	6,728	8,391	10,016	11,770	26.8%
By Segment (EB per Year)							
Consumer	2,997	4,304	5,836	7,435	9,075	10,906	29.5%
Business	1,681	2,218	2,768	3,318	3,853	4,429	21.4%
By Type (EB per Year)							
Cloud data center	3,851	5,636	7,712	9,802	11,850	14,076	29.6%
Traditional data center	827	885	892	951	1,078	1,259	8.8%
Total (EB per Year)							
Total data center traffic	4,678	6,522	8,604	10,753	12,928	15,335	26.8%

Figure 1.1: Global Data Center Traffic Data from *Cisco* (2015)

Fig.1.1 shows the predicted data traffic globally by *Cisco* (2015). Two observations can be easily made from this prediction. First, server-to-server communication demonstrates the fastest growth over this decade. This could be the result of continued rapid growth of cloud and virtualization-based service, such as platform and a service (PaaS) and the trend of distributed solutions, such as the ones deployed in blockchain solutions *Lin and Liao* (2017).

Second, within server communication still dominates the data traffic on the network. This is historically due to the many great ideas from the whole software and

hardware stack that more or less strive to improve the locality and thus better speed, delay, and efficiency throughout *Patterson and Hennessy* (2017). Even in the state-of-art machine learning solutions applied in the real world business, *L'Heureux et al.* (2017) the scale of the solutions are often designed to be within one data center's storage and process capability.

For both within-server communication and between-server communication, wireline transmission schemes are widely deployed, e.g. optical fiber and server backplanes. This is for both historical reasons and the continued advantage of wireline communication in channel condition, energy efficiency and data rate. Meanwhile, mobile communications, i.e. WLAN, LTE, tend to serve server-to-client data traffic due to the much better mobility. In this thesis, we focus on the discussion of wireline communication and thus is more concerned on the problem of massive data traffic itself.

1.1.1 Wireline Communication

Shannon's original work *Shannon* (2001), proposes channel capacity as the ultimate bound for the performance of error-free communications on AWGN channel as in Eqn. 1.1, plotted in Fig. 1.2. Where C is the capacity in bits per second, W, the available bandwidth in the channel in Hz, and SNR the signal-to-noise ratio of the communication system. This classic model divides into two major regions. One is the power-limited region. Most wireless channels fall into this regime due to the high path loss, e.g. LTE systems typically tackle >150dB of loss. The other one is the bandwidth-limited region, where the SNR is usually quite abundant and as shown in Fig. 1.2 increase in SNR provides diminishing gain and thus design focus is generally shifted to the extension of bandwidth. Wireline channels typically belong to this regime, since normally 40dB of loss is considered "the deepest reach" in this market.

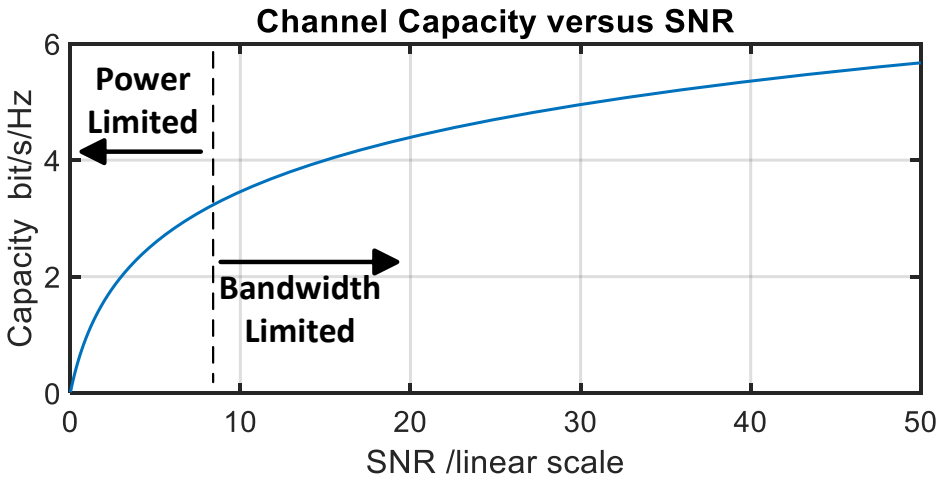


Figure 1.2: Channel Capacity of 1Hz Channel versus SNR

$$C = Wlog_2(1 + SNR) \tag{1.1}$$

1.1.2 The Wireline Channel

Section. 1.1.1 showed the importance of wide bandwidth in high speed wireline communications, from the channel capacity point of view. But unfortunately, copper wires and even optical fibers do not have infinite supply of bandwidth. Even a well-designed transmission line on a PCB board starts to show loss at frequencies as low as tens of megahertz due to skin effect and dielectric loss *Hall and Heck* (2009).

Fig. 1.3 shows a typical channel s-parameter, or the forward transfer function of a wireline channel. The slope of the channel response might not seem intuitively as hostile as white noise, but it does introduce another limiting factor to the performance, known as inter-symbol interference (ISI). ISI can be easily seen on the time domain. Fig. 1.4 shows the response of a single pulse with the same width as one bit period. If such channel were directly used for communication, each bit would leave a residual interference on the channel and thus affecting the detection of succeeding bits.

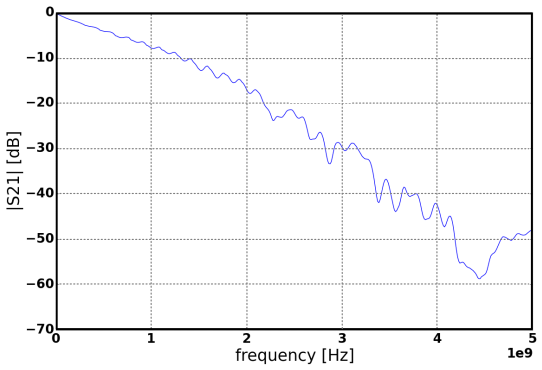


Figure 1.3: Insertion Loss of a Wireline Channel

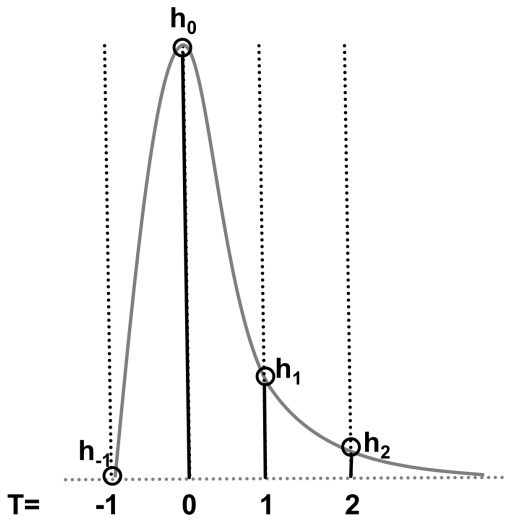


Figure 1.4: Time Domain Output of a Single Pulse

Techniques to expand the bandwidth is the key to removing ISI from wireline channels. Equalization is among the most popular techniques of resolving ISI, and equalization techniques typically trade-off SNR for extra bandwidth, and generally result in a winning situation due to the typically high SNR under wireline channel conditions.

The design of equalization employs analysis on band-limited channels and usually hand-designs and adaptive schemes are employed to ensure a healthy trade-off between SNR and bandwidth.

1.1.3 Equalization Theory on Bandlimited Channel

When analyzing equalization performance one typically considers the abstract model shown in Fig. 1.5. This model ignores timing and serialization which are other two important factors in wireline channel transceiver. Information data stream is typically first encoded with modulation codes, e.g. 8B10B *Zhou et al.* (2017). The coding helps with timing recovery system as well as shaping the data spectrum, cf. *Bergmans* (1996).

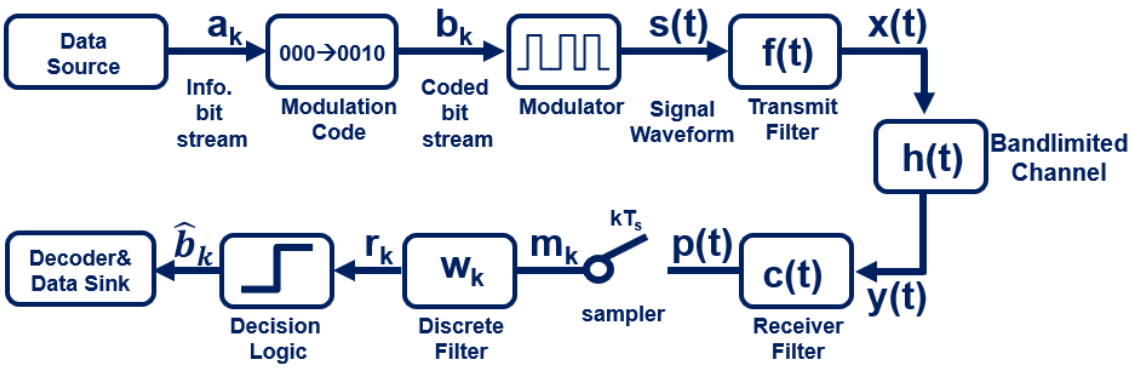


Figure 1.5: Communication system model for equalizer design

Following the modulation code comes the actual equalizer system. Usually two families of equalizers are considered, namely linear equalizers (LE) and nonlinear equalizers(NLE). LEs are filters implemented in either continuous time or discrete time and can be deployed in both transmitters and receivers. They work by either boosting the high frequency component of the signal or suppressing the low frequency part in order to approximate flat response within the signal band. Non-linear equalizers are usually used to fine-tune the equalization output and cancel complex patterns of ISI. Additionally, linear equalizers tend to amplify noise while NLEs are less prone to this problem. The simplest and most commonly used non-linear equalizer is decision feedback equalizer(DFE). A DFE introduces non-linearity by simply subtracting from the received signal a prediction of the trailing ISI using hard decisions and

estimated channel information. Another example of non-linear equalizers is maximum likelihood sequence detection (MLSD) that relies on optimal detection theory. Chapter II discusses MLSD in further detail.

A theoretical result of major interest is the matched filter bound (MFB) under this equalization framework given a band-limited AWGN channel. MFB characterizes the optimal detection performance under ISI. Note that there is still a gap between the performance described by MFB and that of channel capacity, mostly because of the lack of advanced modulation and forward error correction (FEC) in the system under consideration.

Fig. 1.6 shows the intuitive process of deriving MFB. To arrive at an upper-bound of detection under ISI, one considers a lone pulse being sent through the channel. The receiver assumes an optimal matched filter detection to output the polarity of the single lone-pulse. The resulting equivalent SNR is as shown in equation 1.2. Where FSN refers to folded signal-to-noise ratio, which is essentially normalized and scaled spectrum of signal over noise, for more details cf.*Bergmans* (1996).

$$SNR_{MFB} = \int_{-0.5}^{0.5} FSN(e^{j2\pi\Omega}) d\Omega \tag{1.2}$$

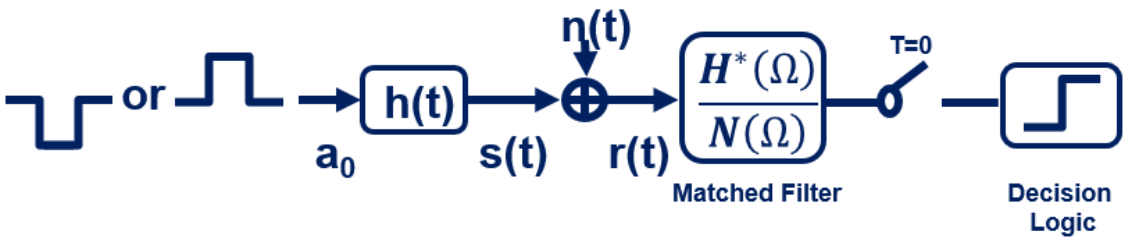


Figure 1.6: Derivation of matched filter bound (MFB)

Another interesting point on the design of equalizers is the theoretical performance comparison among commonly used equalizers and the ultimate upper-bound,

Channel Loss /dB @ Nyquist Frequency	Normalized SNR /dB		
	LE	DFE	MFB
5.8	0	1.1	2.1
8.1	0	1.7	3.4
11.5	0	5.9	10.5

Table 1.1: Comparison of SNR for different equalizers, normalized to LE performance

namely MFB. The comparison boils down to the difference between several average statistics on the FSN *Bergmans* (1996). As shown in Eqn. 1.3 and 1.4, the linear equalizer's performance can be approximated as the harmonic average of FSN while the linear equalizer with DFE can be approximated as the geometric average of the FSN. Numerical evaluations can be generated based on this analysis and real channel data *Palermo* (2017) to quantify the gap as in table 1.1.

$$SNR_{LE} = (\int_{-0.5}^{0.5} \frac{1}{FSN(e^{j2\pi\Omega})} d\Omega)^{-1} \tag{1.3}$$

$$SNR_{DFE} = e^{\int_{-0.5}^{0.5} ln(FSN(e^{j2\pi\Omega}))d\Omega} \tag{1.4}$$

The table shows an expanding gap between LE and DFE, along with between DFE and MFB on the equalization performance. In this thesis, we propose two novel approaches to the equalization problem, both aiming at efficient use of SNR in trading off for bandwidth. The first one is digital signal processing (DSP) oriented, targeting close to MFB performance, while the second one tries to propose a low-power but SNR friendly phase-shaping scheme to boost the SNR of designs under tight power consumption constraint. Analysis on algorithm, architecture and circuit levels will be given in detail in Chapter II and III.

1.2 Security and Cryptography Overview

Security and privacy concerns have always been critical since the beginning of the digital age. As both software and hardware markets are rapidly increasing in volume and branching in varieties, challenges in security issues also come in novel forms.

Examples can be taken from the applications in internet of things (IoT) *Yang et al.* (2017). The fast growth in IoT market has brought forth new digital applications, and meanwhile, threats in security systems have become both more substantial and harder to maintain.

As stated in *Yang et al.* (2017), the number of active IoT devices is projected to be around 41 billion by 2020, straddling applications from personal use like smart home and wearable devices to larger scales like smart grid and vehicle networks. The huge volume and diverse applications have made points of failures in security likely and such failures may cause significant damage to the system or reveal sensitive information of the users. In fact, *Yang et al.* (2017) pointed out that coordinated attacks on smart home systems on a large scale had already been reported, leading to service outage from smart TVs to baby monitors.

Despite its importance, implementation of security schemes for IoT can be quite challenging. Threats can be foreseen at different layers. Especially at IoT perception layer *Yang et al.* (2017), where mobile ends work on data acquisition, the tight constraint on battery life and the light weight computation mechanism deployed can compromise the resource used for crypto-systems infrastructures, leading to potential security failure. Cryptographic schemes such as signatures and key exchange can also be heavily used on other layers, i.e. the network layer and the application layer, and thus light weight and efficient implementations of various crypto-systems is of great interest throughout the IoT market. Meanwhile, IoT only serves as an important example where efficient implementation for security systems can be of substantial value. Other market calling for good security also prevail.

A substantial role as security plays in modern digital applications, new challenges keep emerging and comparable effort has to be dedicated to maintaining the security throughout the whole stack. A recently noteworthy challenge is the fast steps towards the commercialization of quantum computers. As *Mohseni et al.* (2017) pointed out, researchers in quantum computing are expecting a good chance of quantum simulation, sampling and optimization becoming available within a few years from the point of this writing. In fact, on a slightly earlier time, *Howe et al.* (2016), in 2014 large effort has been put in developing quantum attacks on currently active encryption schemes.

In current applications, security systems are mostly deploying primitives including RivestShamirAdleman (RSA), Elliptic Curve Cryptosystems (ECC), Elliptic Curve Digital Signature Algorithm (ECDSA), and Elliptic Curve Diffie-Hellman (ECDH). The applications cover both the Internet *Howe et al.* (2016), and mobile ends *Cheng et al.* (2017). Since the invention of Peter Shor's factorization algorithm and the subsequent improved variants *Monz et al.* (2016), efficient quantum algorithms have been available for integer factorization and (elliptical) discrete logarithm, both of which tend to be deemed hard on classical computation schemes. Since all the crypto-system primitives mentioned above rely on the hardness of these problems, the availability of the first working quantum computer that is complex enough to support the operations in *Monz et al.* (2016) will leave most of the world's network service and devices compromised.

1.2.1 Post-Quantum Security

Fortunately, quantum resilient cryptosystems have been proposed and some are efficient enough to be deployed when necessary. *Howe et al.* (2016) categorized the post-quantum security schemes into four types: *code-based cryptography* that relies on the hardness of decoding a random linear code *Overbeck and Sendrier* (2009),

hash-based cryptography that can be implemented with cryptographic hash functions *Merkle* (1989), *multivariate quadratic cryptography* that uses the hardness of solving quadratic equation sets over finite field *Matsumoto and Imai* (1988), and *lattice based cryptography* that takes advantages of the hardness of geometric problems on lattices *Matsumoto and Imai* (1988).

Lattice based cryptography has recently drawn great attention, due to the promising theoretical hardness results *Regev* (2010), the many recent implementation friendly improvements *Regev* (2010), and the potential for even wider applications in Fully Homomorphic Encryption (FHE) *Khedr et al.* (2016), and in Identity-Based Encryption (IDE) *Ducas et al.* (2014). In Chapter IV we present a lattice encryption instruction accelerator (LEIA) that supports very fast implementations of recent lattice encryption instantiations.

A lattice can be seen as a discrete additive subgroup of $\mathbb{R}^n$ *Lyubashevsky et al.* (2013). Lattice-based cryptography discussions are typically limited to full-rank lattices, ones that can be generated by n linearly independent vectors, as in Def. I.1 and illustrated in Fig. 1.7.

Definition I.1. *Micciancio and Regev* (2009) A **full rank lattice**, Λ is defined as the set of all integer combinations of n linearly independent vectors.

$$\Lambda = \mathcal{L}(\mathbf{b}_1, ..., \mathbf{b}_n) = \{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_i \in \mathbb{Z} \; for \; each \; 1 \leq i \leq n \} \tag{1.5}$$

It is commonly known that algebraic manipulations on lattices tend to be easy and geometric problems on lattices can be very hard. Def. I.2 through 4 gives three well-known problems that are believed to be impossible to solve with polynomial time algorithms, even with the help of quantum computers, cf. *Micciancio and Regev* (2009).

Definition I.2. In **Shortest Vector Problem** (**SVP**), one considers, given a matrix

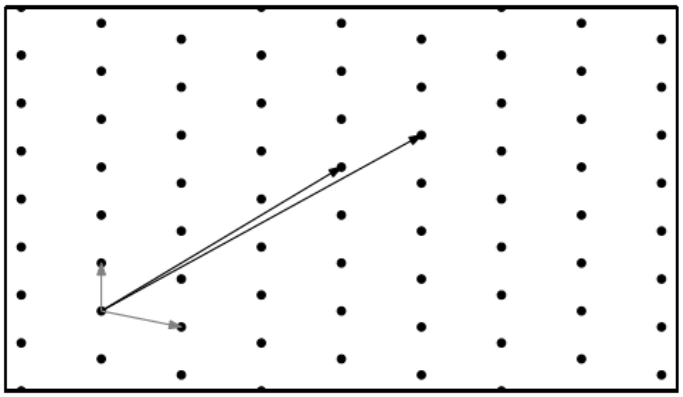


Figure 1.7: An example lattice with two sets of basis vectors *Micciancio and Regev* (2009)

consisting of lattice basis vectors $\mathbf{B}=[\mathbf{b_1}|...|\mathbf{b_n}]$, finding the non-zero vector in $\mathcal{L}(\mathbf{B})$ with the shortest length (in euclidean norm).

Definition I.3. In **Closest Vector Problem** (**CVP**), one considers, given a matrix consisting of lattice basis vectors $\mathbf{B}=[\mathbf{b_1}|...|\mathbf{b_n}]$, and a target vector $\mathbf{t} \in \mathbb{R}^n$, finding the lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ that minimizes the euclidean distance to $\mathbf{t}$, or $\|\mathbf{t}-\mathbf{v}\|_2$.

Definition I.4. In **Shortest Independent Vectors Problem** (**SIVP**), one considers, given a matrix consisting of linearly independent vectors from the lattice $\mathbf{B}=[\mathbf{b_1}|...|\mathbf{b_n}]$, finding another set of n linearly independent lattice points, $\mathbf{B}'=[\mathbf{b_1}'|...|\mathbf{b_n}']$ $\in \mathcal{L}(\mathbf{B})$ that minimizes the length of the longest vector in the basis, $\max_{b_i' \in \mathbf{B}'} \|b_i'\|$.

The construction of cryptosystems typically starts with assumption that certain problems are hard to solve under certain computing resource limitations. Examples include (elliptical) discrete logarithm problems. Effort to construct lattice-based cryptosystems started with Ajtai's seminal work on one-way functions *Ajtai* (1996), and much more work contributed to improving both the security claims and efficiency of implementations.

Among the many lattice-based cryptosystem constructions is the learning with error (LWE) based cryptosystems. LWE is believed to be the most efficient lattice-based cryptosystem with a theoretical support of security *Micciancio and Regev* (2009). The

best known solution to the LWE problems runs in exponential time in n *Regev* (2009), while theory from many other related areas indirectly support the commonly believed conjecture that LWE problems are very hard. For example, solving LWE problems can be shown to simplify to learning parity with noise (LPN) in learning theory and decoding random linear binary codes from coding theory, *Micciancio and Regev* (2009) both are widely believed to be hard already. Moreover, a well-recognized quantum reduction from approximate-SVP and approximate-SIVP to LWE has been established in *Regev* (2009), which indicates that the hardness of LWE can be supported by the hardness of lattice problems even on a quantum computer.

LWE problems can be most simply described in the decision version. Despite the apparent simplicity it enjoys equivalence and hardness proof with many of other seemingly harder problems *Micciancio and Regev* (2009). The decision version of LWE problem, parametrized by integers,n and m, and an often prime integer modulus q, and an error probability distribution χ on $\mathbb{Z}_q$, can be stated as in Def. I.5.

Definition I.5. Given integers m,n,q, and a pair ($\mathbf{A}$,$\mathbf{v}$), where matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ is sampled from uniform distribution and $\mathbf{v} \in \mathbb{Z}^m$, **the decision version of LWE problem** is to distinguish with non-negligible probability of success between the following two cases:

Case I: $\mathbf{v} \in \mathbb{Z}^m$ is sampled from the uniform distribution.

Case II: $\mathbf{v} = \mathbf{A}\mathbf{s} + \mathbf{e}$, where $\mathbf{s} \in \mathbb{Z}^m$ is uniformly chosen and $\mathbf{e} \in \mathbb{Z}^m$ is sampled from the error distribution χ.

Popular choices for the error distribution χ are discrete Gaussian distributions *Micciancio and Walter* (2017) and binomial distributions *Alkim et al.* (2016). Discrete Gaussian (DG) error can be seen as a discrete random variable whose probability mass function (PMF), is the normalized samples from the continuous normal distribution.

Popular approaches in sampling from DG distributions include rejection sampling *Gentry et al.* (2008), discrete Ziggurat *Buchmann et al.* (2013), inversion sampling

Folláth (2014), and Knuth-Yao (KY) sampling *Roy et al.* (2013). Among these methods, KY sampling, which involves random traversal on a discrete distribution generation (DDG) tree, has been recognized to provide the best trade-off between speed and memory usage. More details on DDG tree scheme will be discussed in Chapter IV, and *Folláth* (2014) provides a detailed comparison among these sampling schemes.

Cryptosystems constructed directly on LWE problems tend to require large key sizes, on the order of n^2, affecting both memory and run time performance overhead. A popular approach to improving the efficiency of LWE cryptosystems is to carefully introduce some structure on the LWE samples which both simplifies the computation and reduces the key size, following the heuristic design of NTRU *Hoffstein et al.* (1998). The construction has been generalized to a family of problems called ring-LWE, which has been shown to have promising hardness results, and itself with nice properties, including search-to-decision equivalence *Lyubashevsky et al.* (2010). In the following discussion we follow the construction introduced in *Lyubashevsky et al.* (2013).

In the generalized form of R-LWE *Lyubashevsky et al.* (2013), the operations are defined on the ring obtained by integer polynomials modulo a cyclotomic polynomial, denoted by $\mathbb{Z}[x]/\Phi_m(x)$. A cyclotomic polynomial can be most intuitively explained as the monic minimal polynomial of any primitive m-th root of unity, over the field of rational numbers. It can be analytically defined as in Def. I.6.

Definition I.6. The n-th cyclotomic polynomial, $\Phi_m(x)$, can be defined analytically as

$$\Phi_m(x) = \prod_{\substack{1 \le k \le m \\ gcd(k,m)=1}} e^{(x - j2\pi\frac{k}{n})} \tag{1.6}$$

One convenient and also popular choice of m is any power of 2. In this case $\Phi_m(x) = x^n + 1$, where $n = \frac{m}{2}$. In this case the family of ideal lattices, a structure closely related to the hardness of R-LWE, becomes anti-cyclic lattices, where anti-

cyclic shifts of each vector on the lattice is another vector on the same lattice.

The $\mathbb{R}^n$ representation of elements in $\mathbb{Z}[x]/\Phi_m(x)$ can be straightforward with so called *coefficient embedding*, where each polynomial, $a_n x^n + a_{n-1}x^n + ... + a_0$ is simply represented by a vector $\mathbf{v} \in \mathbb{R}^n$ as $\mathbf{v} = (a_0, ..., a_n)$. However for an arbitrary m, computations, i.e. additions and multiplications modulo $\Phi_m(x)$ can be costly since they will involve convolution and polynomial reductions. A both theoretical and computationally more convenient representation, introduced in *Lyubashevsky et al.* (2013), is *canonical embedding*, Def. I.7, which leads to simple componentwise addition and multiplication on $\mathbb{R}^n$ *or* $\mathbb{C}^n$.

Definition I.7. Let ξ_m be a m-th root of unity whose monic minimal polynomial over the field of rational numbers is the m-th cyclotomic polynomial $\Phi_m(x)$, of order n=$\phi(m)$, where ϕ is Euler's totient function. Then there are exactly n roots for the equation $\Phi_m(x) = 0$ in $\mathbb{C}$, denoted as $x_1, ..., x_n$. Meanwhile there are exactly n ring homomorphisms from $\mathbb{Q}[\xi_m]$ to $\mathbb{C}$, denoted as $\sigma_1, ..., \sigma_n$, with the property that $\sigma_i(\xi_m) = x_i, 1 \leq i \leq n$. The **canonical embedding**, $\sigma : \mathbb{Q}[\xi_m] \to \mathbb{C}^n$ is defined as

$$\sigma(x) = (\sigma_1(x), ..., \sigma_n(x)) \tag{1.7}$$

One very common practice in instantiation of R-LWE is to put m as a power of 2 and thus $\Phi_m(x) = x^n + 1$, where $n = m/2$, and consider all the operations on $R_q = Z_q[x]/\Phi_m(x)$, the ring of integer polynomials with integer coefficients defined modulo q. Also q is considered as a large prime number with the property that $q = 1 \ (mod\ m)$.

Under this common parameter assumption, another computational technique, called number theoretic transform NTT appears to be useful, and can be also easy to operate and store. NTT is essentially an integer version of discrete Fourier transform (DFT), and thus enjoys the property from convolution theorem. Convolutions in the

coefficient embedding domain can be computed after NTT as direct component-wise multiplications. Meanwhile, techniques in designing fast Fourier transform (FFT) can mostly be applied in the NTT and inverse-NTT implementation to give the reduced O(nlog(n)) complexity. Additionally, operating only on integers, NTT is perfectly friendly with fixed-point architectures, that are commonly seen in ASICs and embedded platforms. Def. I.8 gives the analytical definition of NTT and inverse NTT.

Definition I.8. Given prime number q, an integer n such that q = 1 mod n, then there exists a primitive n-th root of unity $\omega_n \in \mathbb{Z}_q$, and its multiplicative inverse ω_n^{-1}. With these assumptions on the parameters, and a sequence $\mathbf{x} \in \mathbb{Z}_q^n$. The **number theoretical transform (NTT)**, and the **inverse number theoretical transform (INTT)** of $\mathbf{x}$ is defined as

$$NTT(x) = \sum_{k=1}^{n} \omega_n^{k-1} \mathbf{x}_k \tag{1.8}$$

$$INTT(x) = n^{-1} \sum_{k=1}^{n} (\omega_n^{-1})^{k-1} \mathbf{x}_k \tag{1.9}$$

The rest of this thesis is organized as follows. Chapter II discusses a maximum likelihood sequence detector powered high speed serial communication transceiver design, both theoretical analysis and experimental results are provided. Chapter III presents another work on optimal equalization that concerns more on the front-end circuit design and proposes a phase equalization scheme (PEQ) that achieves optimal trade-off between intersymbol interference (ISI) and signal-to-noise ratio (SNR). In Chapter IV, a novel lattice encryption accelerator, LEIA, is discussed with detail on the implementations of high performance and energy efficient NTT computation and discrete Gaussian error generation schemes. LEIA achieves almost 4 orders of magnitude superior performance in some of the test cases over the commonly used embedded platforms.

CHAPTER II

A Maximum Likelihood Sequence Detection Enhanced High Speed Serial Link Design

2.1 Introduction

The growing need for data bandwidth is driving the speed requirements of serial peripheral, serial chip-to-chip and serial back-plane communication. State-of-the-art serial link designs are complicated by challenging channel conditions as well as by the non-idealities of deep-submicron analog front-end (AFE) circuits, which are exacerbated at high data rates.

Equalizers are commonly used to compensate for severe channel attenuation and to remove inter-symbol interference (ISI) *Gondi and Razavi* (2007); *Hanumolu et al.* (2005); *Horowitz et al.* (1998); *Liu and Lin* (2004b); *Palermo* (2011). However, the benefits of conventional feed-forward equalizers (FFE) and continuous-time linear equalizers (CTLE) are limited as these amplify noise and degrade the SNR. Decision feedback equalizers (DFE) do not amplify noise, but discard the information stored in pre-cursors and post-cursors from the main cursor leading to suboptimal detection. DFE's hard decision making also results in a loss in soft information and error propagation, causing performance degradation especially when used in conjunction with forward error correction.

MLSD is known as the optimal equalizer for an ISI channel that is subject to Gaussian noise *Forney* (1973). MLSD makes decisions based on a sequence of symbols and their ISI-induced correlations, rather than symbol-by-symbol decisions. Therefore it suppresses failures due to error accumulation and propagation, which hurt conventional DFEs. Moreover, a MLSD does not enhance noise as conventional FFE and CTLE do, and this permits a degraded input SNR, thereby accommodating random noise and random-data-modulated impairments incurred by the AFE.

In various applications requiring detection of digital sequences distorted in a band-limited communication channel or storage media, MLSD has been widely applied to provide low error rate while meeting constrained latency and complexity requirements *Hu et al.* (2017); *Kermani et al.* (2017); *Peng et al.* (2016); *Wang and Kumar* (2017); *Yueksel et al.* (2016a,b). In *Yueksel et al.* (2016a,b), a simplified version of MLSD was implemented and verified on an emulated channel targeting 100Gb/s Ethernet. The authors of *Kermani et al.* (2017) argued that MLSD is also practical for 5-10Gb/s links as it offers a competitive error rate at a reasonable cost of implementation. MLSD and its variants have also been widely present in recent solutions for wireless communications and magnetic storage, e.g., *Hu et al.* (2017); *Peng et al.* (2016); *Wang and Kumar* (2017). However, conventional multi-Gb/s MLSDs consume on the order of 100pJ/b *Anders et al.* (2008); *Bae et al.* (2006); *Black and Meng* (1993); *Elahmadi et al.* (2010); *Veigel et al.* (2013), therefore they have not been reported for use in high-speed electrical serial links.

In this work, we design a new high-speed MLSD architecture for serial links that uses a pipelined look-ahead approach. The new architecture enables sub-10pJ/b optimal equalization. The MLSD is integrated in a high-speed serial link transceiver. The deployment of a full MLSD equalizer enables a low-power design of the AFE to take advantage of the extra error margin by trading off accuracy for a lower cost of power and area. We implement a 5b stochastic flash ADC that reduces both

area and power. An efficient digital clock and timing recovery (CDR) loop is also designed, including a PLL, a Mueller-Muller phase detector (MMPD) and a 32-code phase interpolater. Our key contributions include an efficient, high-speed MLSD architecture inspired by *Fettweis and Meyr* (1989), and the utilization of the extra SNR margin provided by MLSD to tolerate AFE impairments.

The rest of the paper is organized as follows. We first provide a brief overview of the mathematical background of Viterbi algorithm for MLSD in Section 2.2. The serial nature of the Viterbi algorithm makes it challenging to design a high-throughput MLSD. We present a reformulation of the Viterbi algorithm in Section 2.3, based on which efficient and high-throughput look-ahead MLSD can be designed for serial links. A low-power stochastic ADC-based AFE, presented in Section 2.4, takes advantage of the extra margin provided by MLSD to reduce the AFE cost. The MLSD and the AFE are integrated in a prototype 5Gb/s 65nm serial link transceiver. The design of the transceiver and the test chip measurements are summarized in Section 2.5 before the conclusion of this paper.

2.2 MLSD Equalization Theory

Channel distortion places bandwidth limitations that show up as intersymbol interference (ISI) in time domain. With linearity assumptions, which generally holds for channels consisting of passive components, one commonly models the channel as (2.1) *Salehi and Proakis* (2007).

$$\mathbf{y}_i = \sum_{j=-\infty}^{\infty} \mathbf{x}_j * \mathbf{h}_{i-j} + \mathbf{n}_i, \tag{2.1}$$

where $\mathbf{y}_i$'s are observed channel output at the i-th time step from the ADC, $\mathbf{x}_i$'s are the modulated transmitter output, i.e., +1's and -1's in the binary case, $\mathbf{h}_i$'s are the sampled channel response to a single pulse *Hall and Heck* (2009) and $\mathbf{n}_i$'s are the

sampled noise process, usually assumed to be Gaussian. A widely accepted optimal detection technique is MLSD *Forney* (1973) that directly minimizes the probability of error. The MLSD under Gaussian noise and binary transmission assumption takes the form shown in (2.2).

$$\hat{\mathbf{x}} = \underset{\mathbf{x}\in\{+1,-1\}^N}{\arg\min} \sum_{i=-\infty}^{\infty} (\mathbf{y}_i - \sum_{j=-\infty}^{\infty} \mathbf{x}_j * \mathbf{h}_{i-j})^2 \tag{2.2}$$

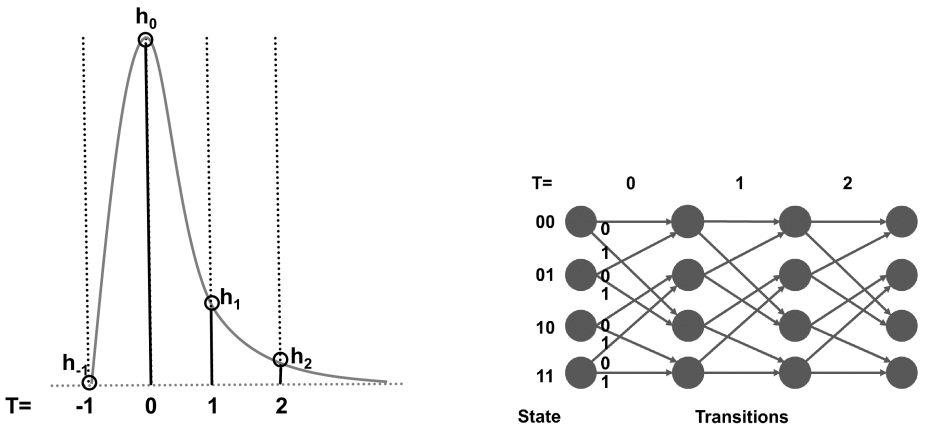
The above estimation essentially minimizes the Euclidean distance between the hypothetical channel response and the actual observation from the ADC. The limits of both sums become finite in reality where both the block length, N and the channel response L become finite or approximately finite. Direct brute force search for a solution to (2.2) would take prohibitive computation cost, on the order of 2^N, where N is the block length.

2.2.1 MLSD and the Shortest-Path Problem

Observing that the channel model can be depicted with a trellis diagram with a fixed and finite number of states and all candidate sequences can be represented as a path through the trellis, the Viterbi algorithm offers a substantially simpler solution that only scales linearly with N.

Fig. 2.1(a) shows an example single-pulse response for a 3-tap channel with one main cursor tap and two post-cursor taps. The constraint length v, is defined as the length of channel memory, e.g., in this case $v = 3$. For a channel of constraint length v, the channel response at a given time point depends on the current bit and also the $v - 1$ bits that are transmitted immediately prior to this time point. For example, the two post-cursor taps shown in Fig. 2.1(a) indicate that the channel response at a given time point depends on not only the bit transmitted at the time point, but also on the two bits transmitted immediately prior to this time point.

In binary signaling applications, there are 2^{v-1} possible combinations of the post-



(a) Example of sampled pulse response of the channel

(b) Corresponding trellis diagram

Figure 2.1: Pulse response and its representation in a trellis diagram.

cursor bits. In a trellis representation, each combination forms a state. Fig. 2.1(b) shows the 4-state trellis diagram corresponding to the 3-tap channel, with the state number labeled on the left and the time steps on top. Note that because of the binary nature of the input data, only two transitions from each state are possible. The temporal adjacency between the bits enforces that every bit sequence corresponds to a state sequence through the trellis, and there exists an explicit one-to-one mapping between the two representations as in Def. II.1.

Definition II.1 (State Sequence)**.** For any binary stream $\mathbf{x}$, $\mathbf{x}_i \in \{-1, +1\}$, modulated on to an ISI channel of constraint length v, one can define a corresponding state sequence $\mathbf{S}$, where $\mathbf{S}_i \in \{-1, +1\}^{v-1}$ is the state at time step i and defined as

$$\mathbf{S}_i = (\mathbf{x}_i, \mathbf{x}_{i-1}, ..., \mathbf{x}_{i-v+2}). \tag{2.3}$$

Note that we assume both sequences are infinitely long at this point for simplicity in indexing. The state at each time step consists of $v - 1$ elements due to channel memory as explained previously. A one-to-one mapping can be easily established in the finite case between $\mathbf{S}$ and $\mathbf{x}$ with the forward mapping defined as in (2.3) and the

inverse defined as

$$\mathbf{x}_i = \mathbf{S}_i[1], \mathbf{x}_{i-1} = \mathbf{S}_{i-1}[1], ..., \tag{2.4}$$

i.e., $\mathbf{x}_i$ as the first element of $\mathbf{S}_i$, $\mathbf{x}_{i-1}$ as the first element of $\mathbf{S}_{i-1}$, etc.

With the state sequence defined, detection of the transmitted bit sequence $\mathbf{x}$ can be equivalently solved as detection of a state sequence $\mathbf{S}$, or a sequence of state transitions. A trellis diagram can thus be described as a graphical representation of all possible state sequences, including the one corresponding to the transmitted sequence. A cost to each state transition, usually referred to as the branch metric, is defined in Def. II.2.

Definition II.2 (Branch Metric)**.** For a state transition at time step i, corresponding to the transition from $\mathbf{S}_i = (\mathbf{x}_i, \mathbf{x}_{i-1}, ..., \mathbf{x}_{i-v+2})$ to $\mathbf{S}_{i+1} = (\mathbf{x}_{i+1}, \mathbf{x}_i, ..., \mathbf{x}_{i-v+3})$, a branch metric $\gamma(\mathbf{S}_i, \mathbf{S}_{i+1})$ is defined as

$$\gamma(\mathbf{S}_i, \mathbf{S}_{i+1}) = (\mathbf{y}_{i+1} - \sum_{j=-\infty}^{\infty} \mathbf{x}_j * \mathbf{h}_{i+1-j})^2, \tag{2.5}$$

which is essentially a term in the summation in (2.2). Therefore the maximum likelihood (ML) optimization (2.2) can be rewritten as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{S}} \sum_{i=-\infty}^{\infty} \gamma(\mathbf{S}_i, \mathbf{S}_{i+1}). \tag{2.6}$$

Observing that each possible state sequence $\mathbf{S}$ is also a path on the trellis diagram, and (2.6) reduces the original ML problem (2.2) to finding the shortest path on the trellis diagram, with the "length" of each step of the path defined as the branch metric on each state transition. The Viterbi algorithm is an efficient way of solving such optimization by utilizing concepts from dynamical programming.

2.2.2 Viterbi Algorithm Formulation

There are two principles underlying the operation of the Viterbi algorithm, namely, the *Principle of Optimality* and the *Principle of Path Convergence*. To use the principle of optimality we need to define path metric, $PM_{i,T}$, as the lowest cost of the state sequence that leads to state T at the time step i, where T is one of 2^{v-1} instantiations of $\mathbf{S}_i$.

$$PM_{i,T} = \min_{\mathbf{S}_i = T} \sum_{k=-\infty}^{i} \gamma(\mathbf{S}_{k-1}, \mathbf{S}_k). \tag{2.7}$$

Direct computation of the path metrics is costly and is almost equivalent to the original ML problem. Given the path metric definition, the *Principle of Optimality*, as stated in Theorem II.3, can be applied to significantly simplify the path metrics computation.

Theorem II.3 (Principle of Optimality)**.** *In the shortest path problem outlined in the previous section, suppose two paths represented by state sequences* $\mathbf{S}$ *and* $\mathbf{S}'$ *intersect at some state* T *at time step* i*, i.e.,* $\mathbf{S}_i = \mathbf{S}'_i = T$*. If*

$$PM_{i-1,\mathbf{S}_{i-1}} + \gamma(\mathbf{S}_{i-1}, T) < PM_{i-1,\mathbf{S'}_{i-1}} + \gamma(\mathbf{S'}_{i-1}, T) \tag{2.8}$$

then $\mathbf{S}'$ *cannot be the sequence corresponding to the shortest path.*

Theorem (Thm). II.3 indicates that if we have the path metrics $PM_{i-1,\mathbf{S}_{i-1}}$, at time step $i-1$, the path metrics for the next time step, $PM_{i,\mathbf{S}_i}$ can be recursively computed utilizing an add-compare-select (ACS) operation, i.e., compute $PM_{i-1,\mathbf{S}_{i-1}}$ + $\gamma(\mathbf{S}_{i-1}, \mathbf{S}_i)$ for all possible transitions from $\mathbf{S}_{i-1}$ to $\mathbf{S}_i$ and select the shortest as $PM_{i,\mathbf{S}_i}$. The *Principle of Optimality* enables the finding of the ML solution in linear time, needing only an initial set of path metrics to start with.

The *Principle of Path Convergence* is an empirical observation that enables further

simplifications of VLSI implementations *Black and Meng* (1993). The *Principle of Path Convergence* states that if we place redundant training vectors of length equal to roughly 6 times the constraint length, v, both at the beginning and at the end of each detection frame, and start ACS operation at the beginning of the leading training vector and decode by tracing back from the end of the trailing training vector, then the decoded output has high probability of converging to the ML solution. Classical Viterbi detectors have all relied on this principle *Black and Meng* (1993).

2.3 High-Throughput MLSD Architecture for Serial Links

One significant limitation on high throughput implementations of Viterbi algorithm is the highly serial nature of the algorithm, so that streams of bits have to be processed one by one. The sliding block architecture *Bae et al.* (2006); *Black and Meng* (1993) has been a popular approach to speeding up the design. However, it suffers from a pre-training overhead of about $6v$ on each side. The overhead also includes both a widened deserializer block and deep skew buffers.

2.3.1 Matrix Formulation of Viterbi Algorithm

To arrive at a more efficient high-throughput architecture, we look at the matrix formulation of the Viterbi algorithm *Fettweis and Meyr* (1989). From the trellis diagram, we can define a $2^{v-1} \times 1$ cost vector $\mathbf{C}_i$ by grouping the path metrics of all the 2^{v-1} states at time step i of the trellis. For the edges between time step $i-1$ and i we can define a $2^{v-1} \times 2^{v-1}$ transition matrix $\mathbf{M}_i$.

To facilitate the mathematic formulation we also define operations $\boxplus$ and $\boxtimes$, both on real numbers as in Def. II.4.

Definition II.4 (Add and Multiply Operations)**.** On the real numbers, $a, b \in \mathbb{R}$, a

pair of add and multiply operations can be defined as

$$a \boxplus b = min(a, b) \qquad (Add) \tag{2.9}$$

$$a \boxtimes b = a + b \qquad (Multiply). \tag{2.10}$$

It can be shown that the set of real numbers together with these two operations form a semi-ring *Fettweis and Meyr* (1989), which essentially justifies the use of basic matrix manipulations. Now with all these concepts defined it can be easily shown that the ACS operations can be seen as

$$\mathbf{C}_i = \mathbf{M}_i \mathbf{C}_{i-1}. \tag{2.11}$$

The original Viterbi algorithm can be understood as simply starting with an initial cost vector and sequentially multiplying the transition matrices with the cost vector. One can invoke the associative law to group the product of the transition matrices and rewrite (2.11) as

$$\mathbf{C}_N = \Big(\prod_{i=-\infty}^{N} \mathbf{M}_i \Big) \mathbf{C}_0, \tag{2.12}$$

where $\mathbf{C}_0$ is the initial condition for the path metrics.

Since the ACS operations are equivalent to the matrix-vector multiplication based on the foregoing discussion, a generalized ACS operation is equivalent to the matrix-matrix multiplication.

Theorem II.5 (Generalized Principle of Optimality)**.** *Suppose two state sequences* $\mathbf{S}$ *and* $\mathbf{S}'$ *intersects at two states,* T *at time step* i *and* U *at time step* j*, i.e.,* $\mathbf{S}_i = \mathbf{S}'_i = T$*, and* $\mathbf{S}_j = \mathbf{S}'_j = U$*, and assuming* $i < j$ *without loss of generality. If*

$$\sum_{k=i}^{j-1} \gamma(\mathbf{S}_k, \mathbf{S}_{k+1}) < \sum_{k=i}^{j-1} \gamma(\mathbf{S'}_k, \mathbf{S'}_{k+1}) \tag{2.13}$$

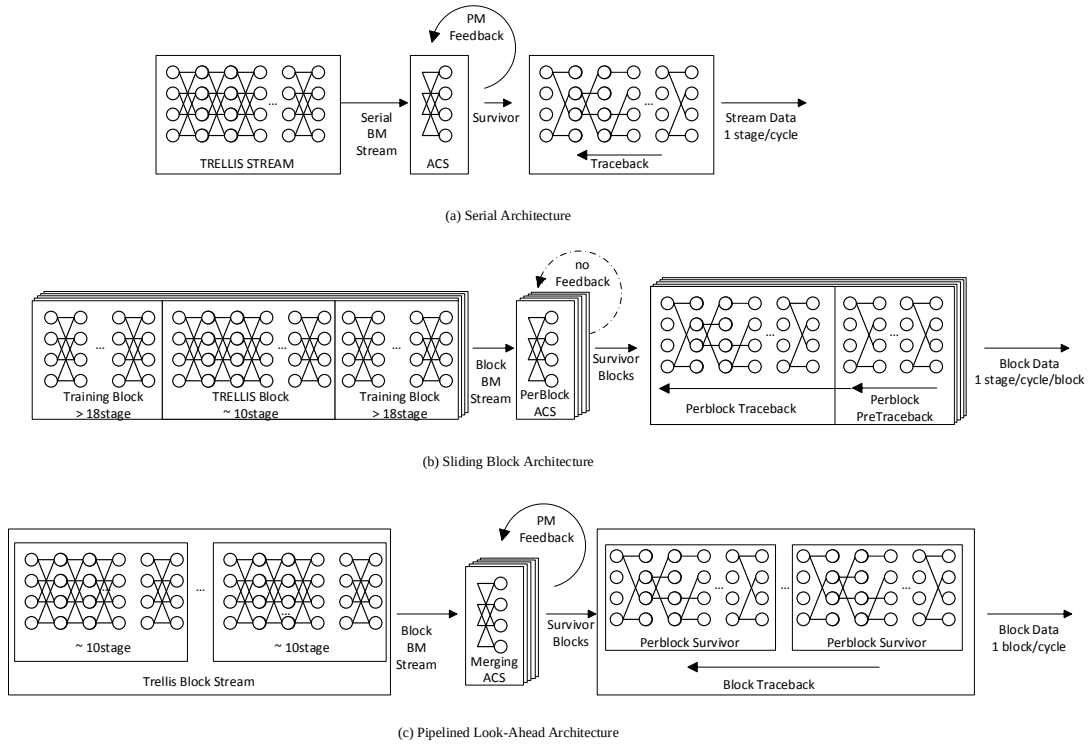


Figure 2.2: MLSD architectures: (a) serial architecture; (b) sliding block architecture; and (c) pipelined look-ahead architecture.

then $\mathbf{S}'$ *cannot be the ML solution sequence.*

The matrix-matrix multiplication used in (2.12) is just a result of the direct application of Thm. II.5. This principle can be viewed as a generalization of the *Principle of Optimality* discussed previously as well as a reformulation of the matrix form of the Viterbi algorithm in (2.12). In the following section, we discuss our implementation based on the *Generalized Principle of Optimality*.

2.3.2 Efficient and High-Throughput MLSD Architecture

A serial MLSD architecture based on (2.11) is illustrated in Fig. 2.2(a). This architecture computes path metrics $\mathbf{C}_i = \mathbf{M}_i\mathbf{C}_{i-1}$, one stage at a time. Due to the recursive nature of the computation, i.e., each stage requiring the path metrics

from the previous stage, the latency of the serial architecture is $\mathscr{O}(N)$. Since $\mathbf{M}_i$ is a $2^{v-1} \times 2^{v-1}$ matrix and $\mathbf{C}_{i-1}$ is a $2^{v-1} \times 1$ vector, one stage of this architecture requires 2^{v-1} ACS units. One major drawback of the conventional serial architecture is that it is impossible to apply look-ahead to this architecture to speed up the computation. The path metrics must be computed one stage at a time due to data dependency, which severely limits the throughput of this architecture.

A popular approach to breaking the throughput bottleneck of the serial architecture while still using (2.11) is shown in Fig. 2.2(b). By dividing data into blocks and concatenating training frames at the beginning and end of each block, the data dependency between blocks becomes approximately negligible. Thus the data processing can be highly parallelized or deeply pipelined to speed up the operation *Black and Meng* (1993). However, the sliding block architecture requires a long pre-training frame on each side, and it can become an exessive overhead. In the example shown in Fig. 2.2(b) with a constraint length of $v = 3$, 36 training stages are required in total to decode a block.

Inspired by the M-step algorithm*Fettweis and Meyr* (1989), we propose an alternative serial MLSD architecture based on (2.12) to overcome the deficiencies of the serial architecture and the sliding block architecture. This architecture "combines" transition matrices, $\mathbf{M}_i\mathbf{M}_{i+1}$, one pair at a time. Compared to the conventional serial architecture that performs one matrix-vector multiplication at a time, the alternative serial architecture performs one matrix-matrix multiplication at a time, which is more expensive. The transition matrices are $2^{v-1} \times 2^{v-1}$, so a stage of this architecture requires $2^{2(v-1)}$ ACS units. A key feature of this alternative serial architecture is that it can proceed without requiring the path metrics, eliminating data dependency.

The lack of data dependency makes it possible to apply look-ahead by combining transition matrices through parallel or pipeline approaches. The combining of transition matrices can be done independently without waiting for path metrics, enabling

a significant improvement in throughput. A P-stage pipelined or P-stage-parallel implementation of this look-ahead architecture is capable of combining P transition matrices in every clock cycle after an initial latency of P clock cycles for the pipelined implementation or $\log_2 P$ for the parallel implementation. A 10-stage look-ahead architecture is illustrated in Fig. 2.2(c). An important difference between the look-ahead architecture and the sliding block architecture is that no pre-training frames are needed for the look-ahead architecture, resulting in a much lower hardware complexity and thus a much higher efficiency.At our design point, with P=10, the pipeline look-ahead approach saves about 20% on the number of ACS, and 90% on the skew buffering, and thus resulting in much higher energy efficiency.

Comparing the pipeline and parallel implementation options of the look-ahead architecture, the parallel look-ahead architecture has a lower latency if P is relatively large, but it also requires P to be a power of 2 to fit an ideal binary tree structure. The pipelined look-ahead architecture incurs a higher latency but it imposes no requirements on P.

For a multi-GSample/s (GS/s) serial link application, it is only feasible for the digital equalizer to run at a fraction of the sample rate. A P-stage pipelined or P-stage parallel look-ahead MLSD is capable of combining P stages of transitions matrices in one clock cycle, allowing the digital equalizer to run at a clock frequency of f_s/P, where f_s is the sample rate. For flexiblity in choosing P and not being bound by the power of 2 requirement, we use a pipelined look-ahead architecture in implementing the MLSD.

2.3.3 MLSD Implementation for Serial Links

The design of the MLSD is a tradeoff between robustness and complexity. A detector with more taps, i.e., v, offers a wider timing margin, but the complexity of the MLSD scales exponentially with v as discussed in the previous section. Our

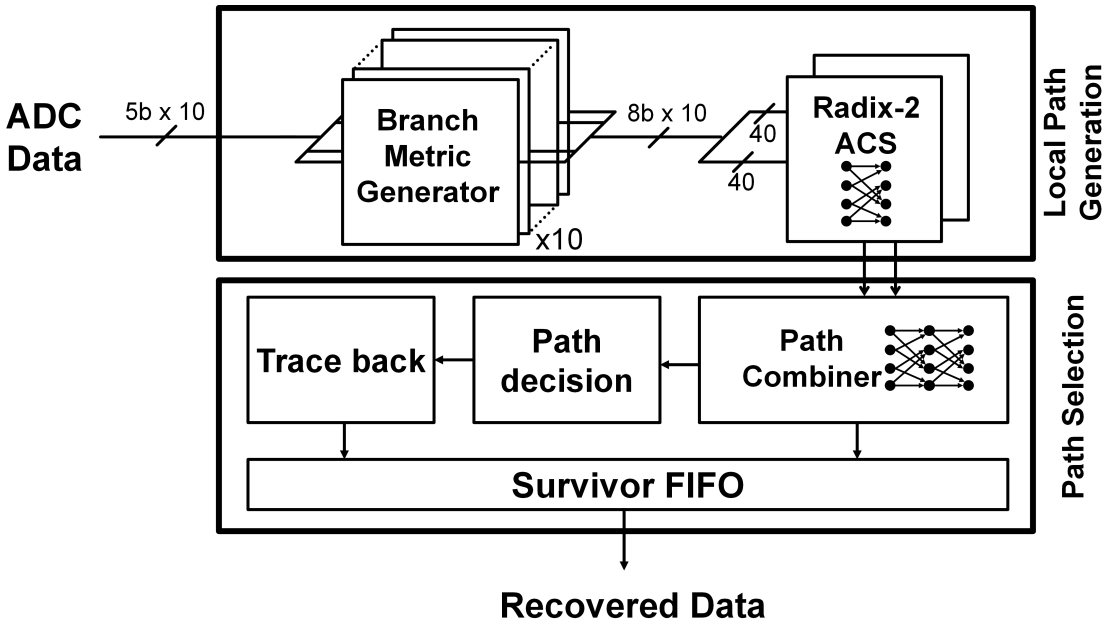


Figure 2.3: Pipelined look-ahead MLSD implementation.

simulation shows that a 4-tap MLSD running at 5Gb/s provides a 24ps timing margin at 10^{-8} BER on a channel with 21dB loss at Nyquist rate; and a 3-tap detector narrows the margin to 12ps at 10^{-8} BER under the same setup, but still sufficient for our application. Therefore the 3-tap MLSD is chosen for our design. The taps of MLSD can also be reprogrammed or adapted to accommodate different data rates and loss.

Given a target 5GS/s serial link application, the samples need to be deserialized to be processed by the MLSD. In a 65nm technology, the digital MSLD can be designed to run at a 500MHz to 1GHz clock frequency. Given the sampling rate and the target clock frequency for the MLSD, we choose $P = 10$ and design a 10-stage pipelined MLSD as shown in Fig. 2.3.3. In each clock cycle, the MLSD collects a block of 10 5b samples to compute 10 branch metrics in parallel.The branch metric calculation is done using a 5b lookup table to provide the flexibility in optimally programming the branch metric. In our design, we programmed the lookup table based on scaled Euclidean distance combined with minor correction of the AFE.

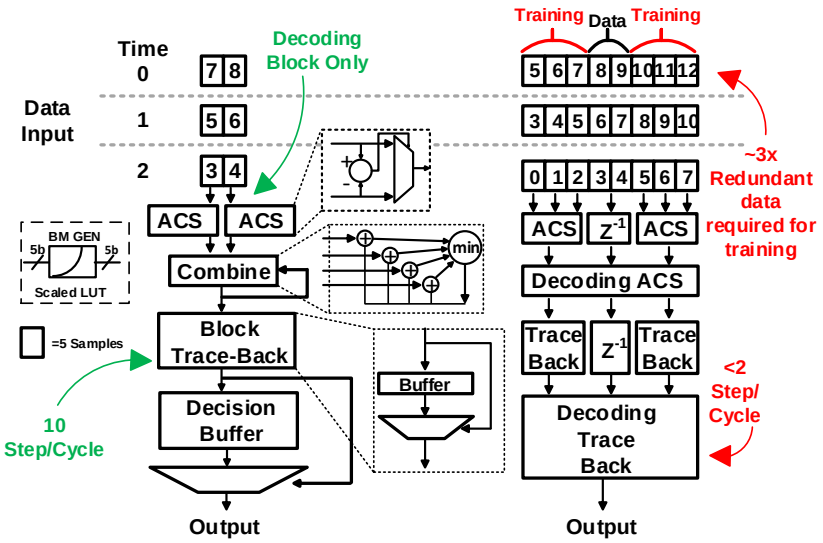


Figure 2.4: Comparison of the pipelined look-ahead architecture (left) and the sliding block architecture (right).

The 10-stage pipelined design is further structured in two 5-stage pipeline parts, one part combining transition matrices forward from stage 1 to 5; and the other part combining backward from stage 10 to 6. Compared to a standard 10-stage pipelined design, the bi-directional design reduces the number of skewing buffers. As illustrated in Fig. 2.3.3, 10 branch metrics are fed to two sets of ACS units (one forward and one backward) to successively compute the transition matrix in a 5-stage pipeline. Forward and backward operations produce two transition matrices after a latency of 5 clock cycles. A final combiner combines the two transition matrices and keeps the survivor path. Each survivor path is buffered and accumulated for 3 blocks until a path decision is made, which is equivalent to a 30b trace back in a conventional Viterbi detector.

After an initial latency of 15 cycles, this pipelined look-ahead architecture is capable to process 10 trellis stages per clock cycle. Assuming binary signaling, the MLSD outputs 10 bit per clock cycle, i.e., 5Gb/s at a 500MHz clock frequency, or 10Gb/s at 1GHz.

We use several methods to improve power efficiency of the MLSD, as illustrated in Fig. 2.4. First, multiplications in calculating branch metrics are replaced by simple lookup tables with 5b precomputed scaled Euclidean distances. Second, the pipelined look-ahead architecture eliminates redundant training calculations necessitated by fine-granulated ACS and trace-back blocks in the sliding-block architecture. Our new approach retains all the confidence metrics from the past sample blocks instead of relying on training, and thus conceptually suffers much less from the well-known edge effects that usually occur in conventional MLSD designs. A well-known high-speed MLSD design based on *Black and Meng* (1993) is also shown in Fig. 2.4 for comparison. To achieve the same 5Gb/s throughput, our pipelined look-ahead architecture saves 75% of buffering and computation in ACS, and incurs 75% shorter latency in traceback.

2.4 Analog Frontend Implementation

The robustness of the MLSD facilitates an energy-efficient AFE architecture, which utilizes an efficient interleaved stochastic flash ADC and a digitally controlled clock recovery loop. Furthermore, since the MLSD creates more margin for AFE non-idealities, there is no need for a front-end equalizer or input buffer. Our system analyses based on methods provided in texts like *Hall and Heck* (2009) have shown that MLSD provides more than 4dB of extra SNR with 10dB loss at Nyquist rate, which significantly eases both the timing and offset of the AFE. These approaches all contribute to a higher power efficiency and a simple AFE design.

2.4.1 Stochastic Flash ADC Design

The power, area and input capacitance of the ADCs are bottlenecks in ADC-based serial-links. In this design, we use a stochastic flash ADC *Pernillo and Flynn* (2011) to keep the power consumption and input capacitance low compared to con-

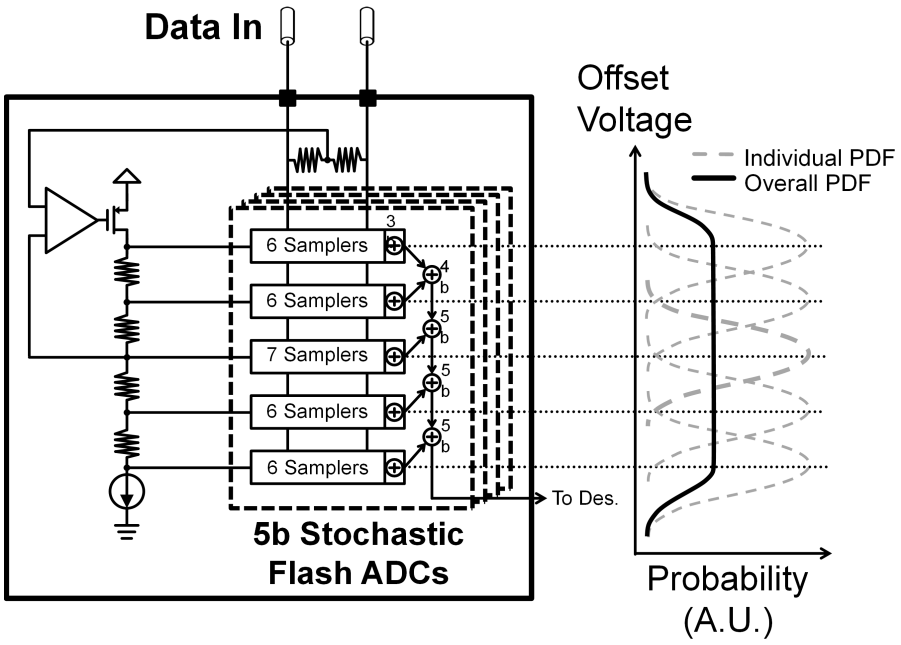
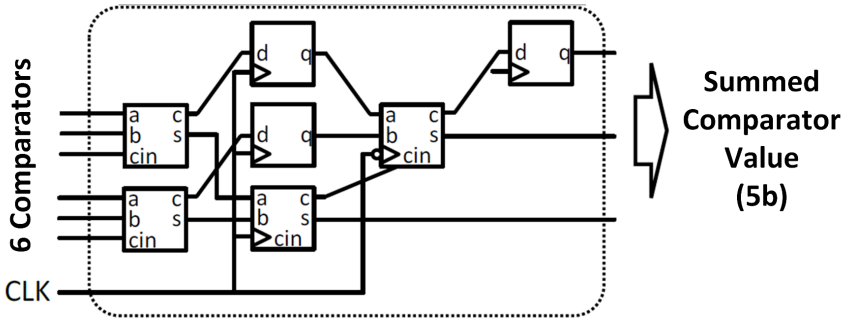


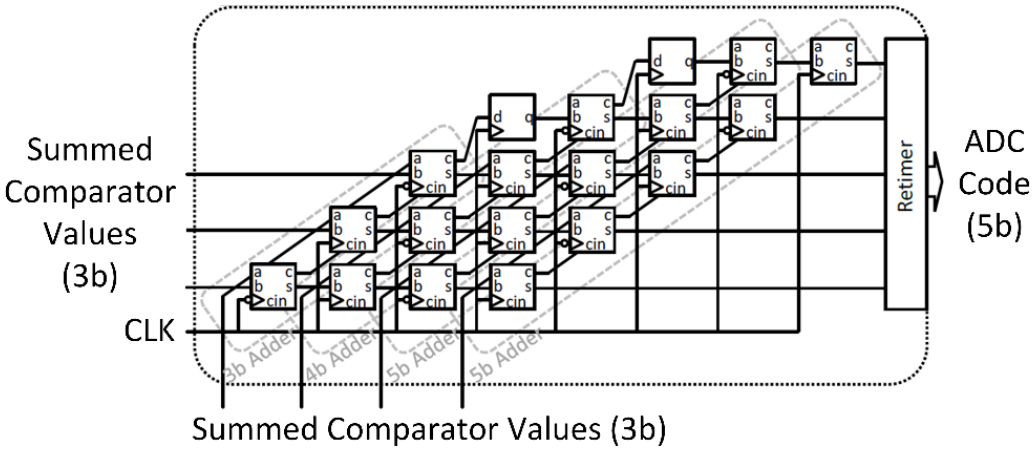
Figure 2.5: Stochastic flash ADC design.

ventional flash ADCs *Cao et al.* (2010); *Ting et al.* (2013). The 2x-oversampled and 4-way interleaved ADC shown in Fig. 2.5 utilizes the random Gaussian input offset distribution of small comparators to collectively give a near-uniform distribution of comparator trip voltages across the input signal range (nominally 200mV differential peak-to-peak).

Instead of the conventional flash-ADC array of accurate comparators driven off a power-hungry reference ladder, we exploit the large, and normally undesired, offsets of small efficient comparators to set the trip points of the ADC. StrongArm comparators are fast, and give reasonably large random offsets. For the ADC to cover the full signal range, the comparators are grouped and tied to different coarse reference voltages taken from a low-power resistor string. This effectively spreads out the individual random offsets, to evenly distribute the ADC trip points across the entire signal range. Low threshold voltage (LVT) devices are extensively used in the first stage of the comparator, to make it run at higher speed.



(a) Adder structure for local summation



(b) Adder structure for final summation

Figure 2.6: Adder structures used in stochastic ADC.

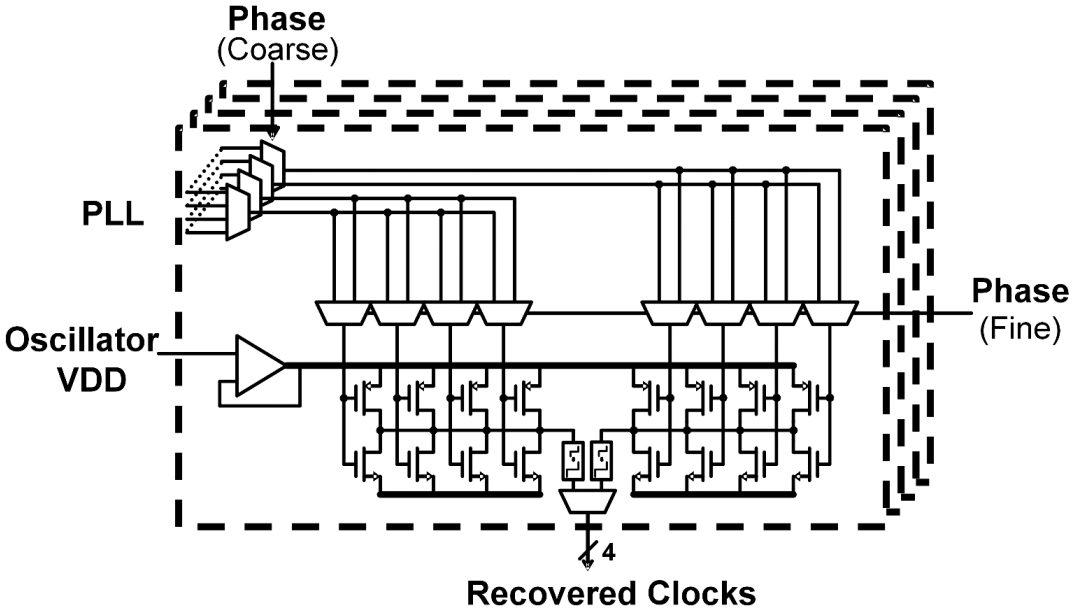


Figure 2.7: Digital phase rotator design.

The ADC uses an adder as an encoder because the trip points are scrambled across the range. The outputs from every group of comparators (6 or 7 comparators) are summed locally to yield 3b values, as shown in Fig. 2.6(a). These values are then summed using structure shown in Fig. 2.6(b) to give the overall 5b ADC code. Four interleaved ADCs sampling at 2.5GS/s deliver an aggregate sampling rate of 10GS/s, oversampling by a factor of 2 to facilitate timing recovery. Small time-interleaving errors between the four ADCs, as well as ADC non-idealities are modulated by random data input, resulting in random errors that are partially corrected in the branch metric lookup table while the rest are tolerated by the MLSD's error margin.

2.4.2 Phase and Timing Control

The extra margin and robustness of the MLSD also permits the use of a compact and efficient inverter-based digitally controlled phase rotator, shown in Fig. 2.7, that generates 32 phases from the 8 phases produced by the on-chip PLL.

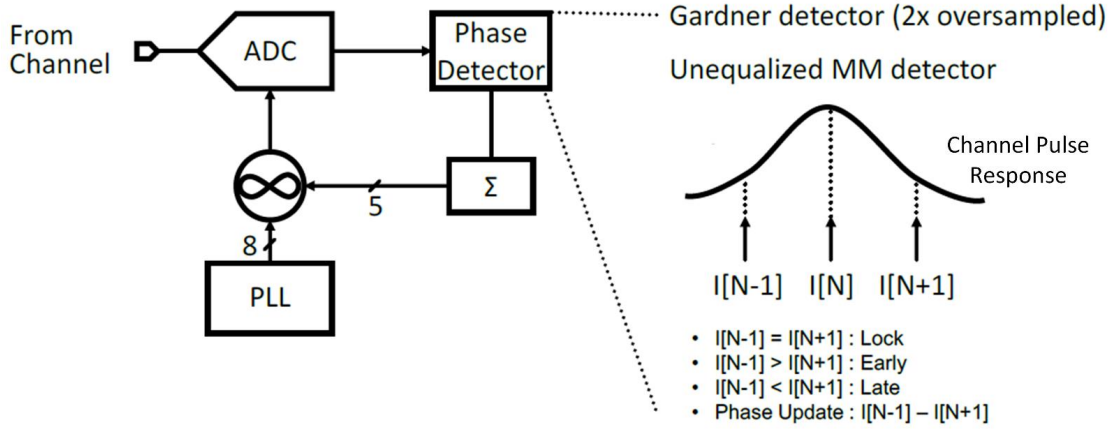


Figure 2.8: Clock recovery loop with unequalized Mueller-Muller detector.

The digital clock recovery loop selects the phase that best represents the center of the unit interval (UI), as shown in Fig. 2.8. For better linearity of interpolation, the oscillator VDD from the rail of PLL VCO sets the supply voltage for the inverters in the interpolator to adjust the slope of the internal signals for the clock rate. In this way, the slope of the internal interpolation signal extends over two adjacent input phases so that the interpolator operates in a more linear fashion.

The phase detector takes the ADC samples and performs early/late detection and loops the information back into the phase rotator via an accumulator. The system is first-order, thus is unconditionally stable. The un-equalized Mueller-Muller phase detector (MMPD) implements the standard MMPD logic, $d_{k-1}y_k - d_k y_{k-1}$, where d's are decisions from MLSD and y's are ADC samples. The CDR takes the derivative of the input data stream to generate the impulse response and detects whether the pre-cursor and the post-cursor are balanced with each other. For successful phase detection, the impulse response has to extend across multiple sampling intervals, which has to be guaranteed by the channel.

Table 2.1: Comparison with Previous ADC-Based Work

	This Work	*Chen et al.* (2012)	*Cao et al.* (2010)	*Ting et al.* (2013)	*Zhang et al.* (2013)	*Shafik et al.* (2016)
CMOS Tech.	**65nm**	65nm	65nm	65nm	40nm	65nm
Data Rate	**5Gbps**	10Gbps	10.3Gbps	10Gbps	8.5-11.5Gbps	10Gbps
Channel Loss	**21dB @2.5GHz**	29dB @5GHz	26dB @5GHz	10dB @5GHz	34dB @5GHz	36.4dB @5GHz
ADC Type	2x2-Way Interleaved Stochastic Flash	4-Way Interleaved Non-Linear Flash	4-Way Interleaved Flash	4-Way Interleaved Flash	4-Way Interleaved Rectified Flash	32-Way Ti-SAR
ADC Res.	**5b**	4b	6b	5b	4b	6b
AFE Power (mW)	**24.7**	63	500	110	195	79
AFE Area (mm^2)	**0.14**	0.185	3	0.25	0.82	0.38
EQ Structure	**3-tap Viterbi**	5-Tap DFE	FFE+DFE	2-Tap DFE	FFE+DFE	FFE+DFE
EQ Power (mW)	**19.3**	37	-	111	-	10
EQ Area (mm^2)	**0.21**	0.075	-	0.286	-	0.39
Total Power (AFE+EQ) (mW)	**44**	130	500	221	160	89
Energy Efficiency (pJ/bit)	**10.9**	13	48.5	22.1	18.9	8.9
Core Area (mm^2)	**0.35**	0.26	15	0.536	-	0.81

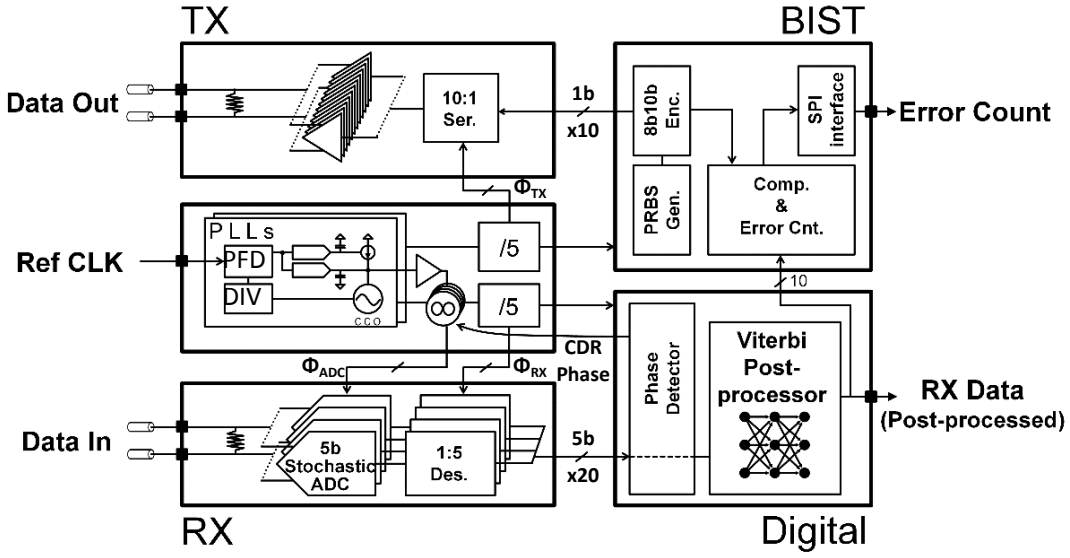


Figure 2.9: Transceiver architecture.

2.5 Prototype Design and Measurements

The overall architecture of our prototype 5Gb/s ADC-based serial-link transceiver with MLSD is shown in Fig. 2.9. The prototype includes transmitter, receiver, on-chip clock generation and timing recovery and the digital MLSD. To facilitate testing, the prototype incorporates a pseudo-random bit sequence (PRBS) data generator and a bit error counter.

2.5.1 Design Summary

We exploit the robustness of the MLSD to simplify the AFE and remove the need for a power-hungry analog equalizer. For energy efficiency, area efficiency and speed, a 10GS/s and 4-way interleaved, 5b stochastic flash ADC 2x oversamples the input signal. The clock recovery loop is closed by a bang-bang phase detector which extracts and integrates phase information from the ISI-corrupted data sampled by the ADCs. A digital phase-rotator finely adjusts the sampling clock phases.

The ADC outputs are de-serialized to form blocks of 10 to be processed at 500MHz by the MLSD, which decides the most probable bit sequence. The prototype also incorporates a 5Gb/s transmitter, a PRBS data generator and a bit error rate tester.

The prototype 5Gb/s transceiver is fabricated in 65nm GP CMOS and packaged in a QFN60 package. The chip microphotograph is shown in Fig. 2.10. The complete transceiver system occupies an area of 700μm x 1400μm and the MLSD takes only 700μm x 300μm.

2.5.2 Measurement Results

The MLSD is evaluated at 500MHz to achieve 5Gb/s at a 750mV supply, dissipating a measured 19.3mW. At 1.0V, the MLSD is evaluated at 1GHz to achieve 10Gb/s, dissipating 57.9mW. The energy-per-bit FoM, defined by the average energy consumption for receiving one bit, is compared to the prior art in Fig. 2.11.

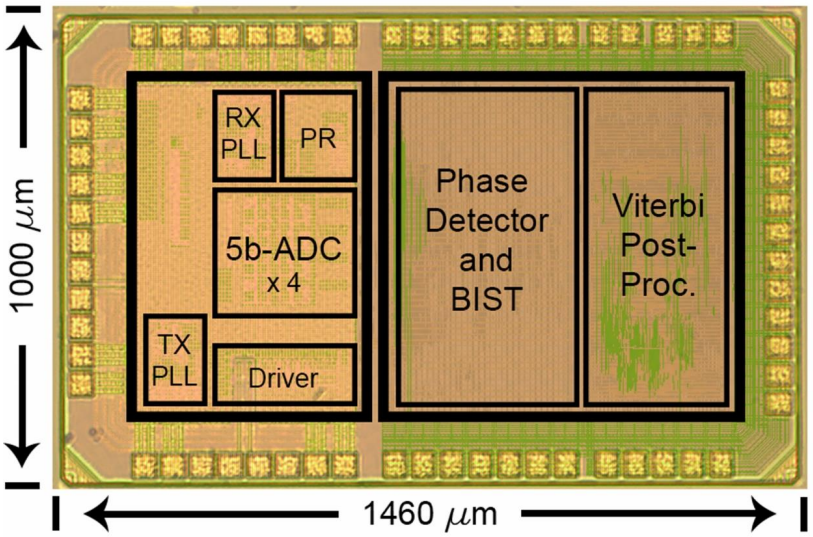


Figure 2.10: Chip microphotograph.

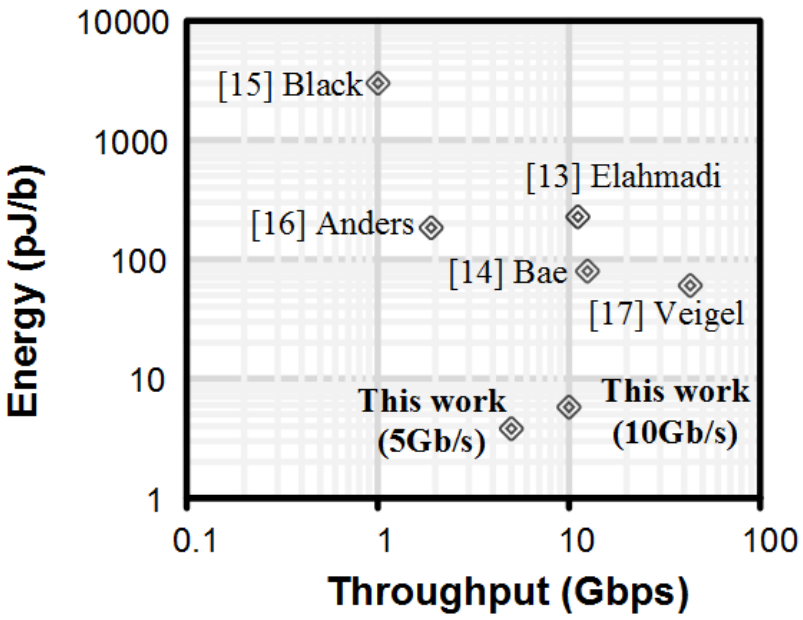


Figure 2.11: Comparison of multi-Gb/s MLSD implementations.

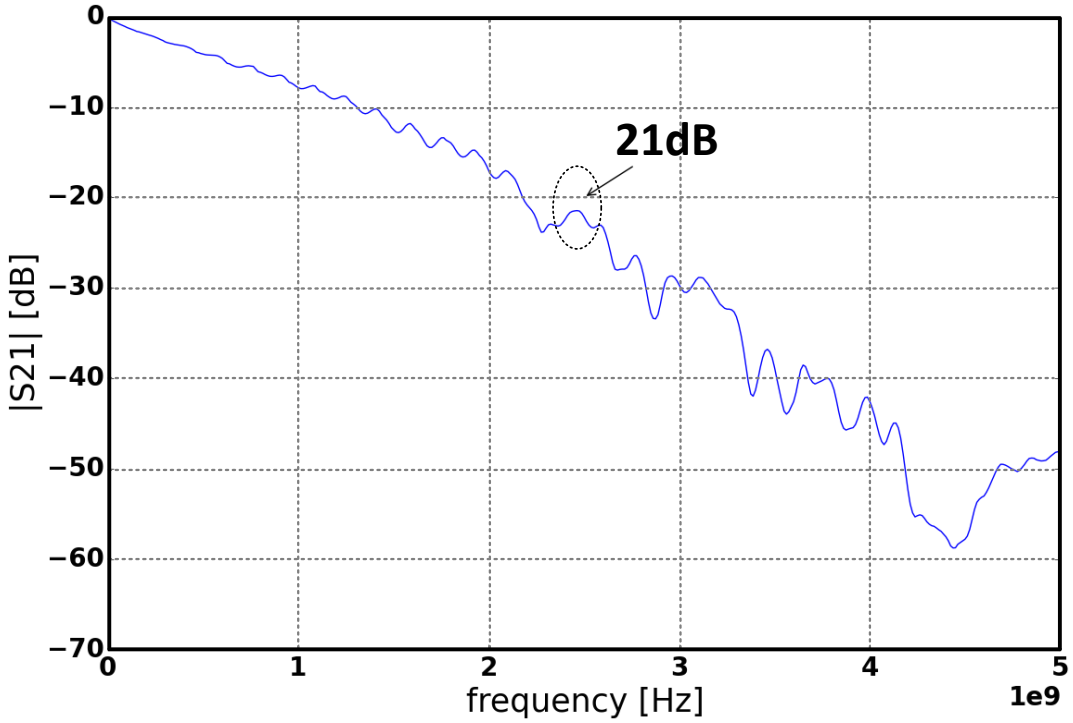


Figure 2.12: Insertion loss of the test FR-4 trace.

The 5Gb/s MLSD is the smallest among all the previously published MLSDs for link applications, and it improves the energy efficiency by more than an order of magnitude. All the MLSDs are compared against the 3-tap MLSD or running using a 3-tap configuration.

Transmit and receive operation are verified at 5Gb/s. The TX is implemented with programmable 112 unit drivers and a pattern generator for full coverage of test patterns. The chip incorporates built-in self-test to monitor the BER of the transceiver. For BER testing, a PRBS-31 sequence, encoded by 8b10b, is sent by the transceiver over a 45cm FR-4 trace. The channel has a measured attenuation of 21dB at 2.5GHz (Fig. 2.12) and testing shows that BER under this condition is better than 10^{-11}. The measured bathtub curve is shown in Fig. 2.13.

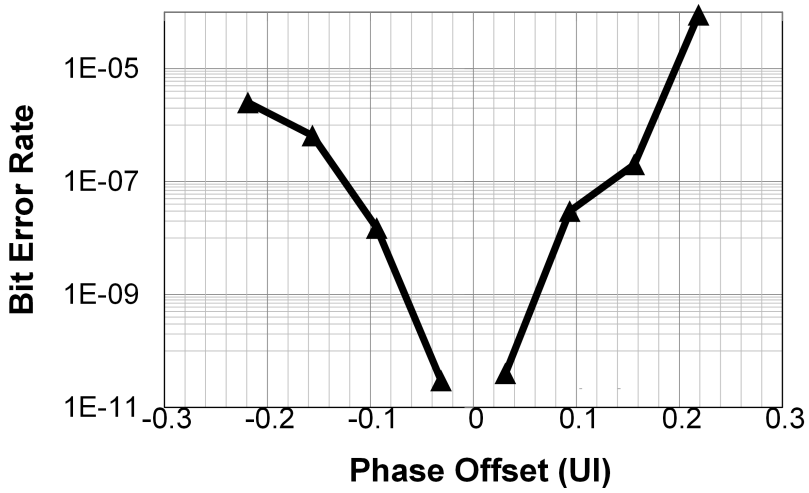


Figure 2.13: Test setup bathtub curve.

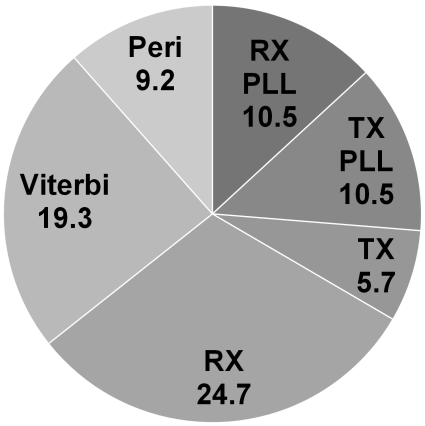


Figure 2.14: Test chip power measurements (mW).

The power breakdown is presented in Fig. 2.14. The total power consumed by the receiver is 54.5mW (with PLL) at 5Gb/s with a 950mV AFE supply and a 750mV digital back end supply. The peripheral power includes front-end BIST, SPI interface and BER tester. The entire receiver FoM is 10.9mW/Gb/s.

Performance metrics are summarized and compared to state-of-the-art ADC-based serial link designs *Cao et al.* (2010); *Chen et al.* (2012); *Shafik et al.* (2016); *Ting et al.* (2013); *Zhang et al.* (2013) in Table. 2.1. For similar channel loss and data rate, our design demonstrates competitive power, area and energy efficiency. Furthermore, as presented in the sections above, by deploying a MLSD, our design also enjoys a large margin for compatibility with different applications and relaxed timing and noise constraints on the AFE.

2.6 Conclusion

In this work, we present a new pipelined look-ahead MLSD architecture for serial links. The architecture provides a high throughput, up to 10Gb/s, and eliminates the pre-training overhead of the conventional sliding block architecture to achieve an efficiency of 5.79pJ/b in a 65nm test chip design. The efficiency exeeds the state-of-the-art multi-Gb/s MLSDs by over an order of mangitude.

Utilizing the extra timing and noise margin provided by the MLSD, we designed a serial link transceiver using a 5b stochastic flash ADC and a digitally controlled clock and timing recovery loop. The complete 65nm transceiver chip was verified at a BER of 10^{-11} on a 45cm FR-4 trace, with 21dB loss at Nyquist frequency. Including all test structures, the chip occupies only $0.88mm^2$. The design achieves a competitive FoM of 10.9mW/Gb/s.

CHAPTER III

A Phase Equalization Enhanced Wireline Transceiver

3.1 Introduction

Wireline channels usually involve two types of hard constraints. The first one is the channel loss. As the data speed goes up, the loss at Nyquist frequency tends to degrade, putting more load on the equalizers, as well as lowering the upper bound on receiver performance, i.e. matched filter bound. Chapter II discusses the optimal detection based scheme that can come in handy when the channel loss dominates design constraints.

A second constraint is the intrinsic limitation on the circuit. Todays communication systems are heavily implemented in CMOS integrated circuits. Even though CMOS transistors can provide transit frequency near 200 GHz, digital implementations usually exhibit sub-optimal trade-off between power and performance in the high frequency regime. The extra power consumption could be partially explained by the requirement of maintaining square wave outputs throughout the system. Therefore, under tight power consumption and area budget, digital-heavy implementations as discussed in II are unfortunately not always feasible.

State-of-the-art link designs are subject to both constraints, and thus much room

for innovation is present for optimal equalization while that consumes minimal power and hardware resource.

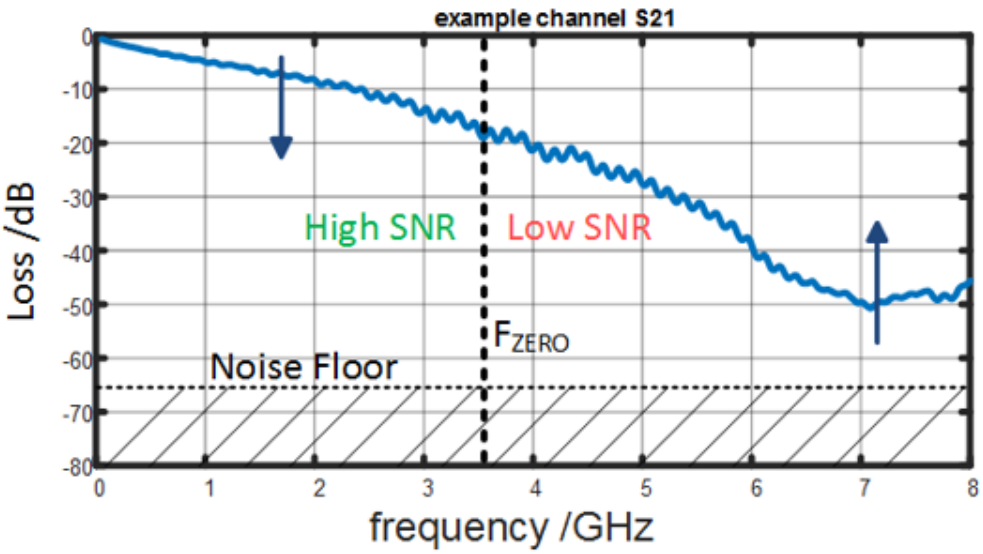
In this chapter, we discuss the design of a phase equalization scheme that provides a novel towards achieving the optimal trade-off between SNR and bandwidth in baseband communications and the solution is optimally positioned in the low power and high performance regime.

3.2 Noise Analysis in Conventional Equalizer

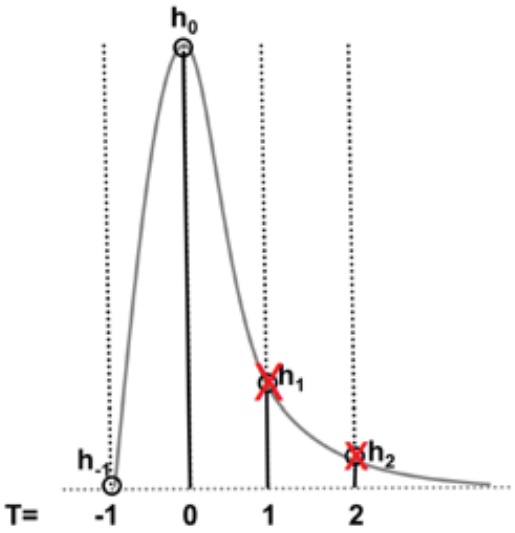
Due to the simplicity of implementation and thus lower power consumption, linear filter based designs are almost universally used in high speed baseband communication systems. However, almost all designs under this regime would not be able to eliminate either ISI or noise completely but rather there usually exists a trade-off between the residual noise and residual ISI. As shown in Fig. 3.1.

A CTLE reduces ISI by either boosting the high-frequency component or canceling the low-frequency component. But this would degrade the SNR since the channel filters out the high-frequency component of the signal making the SNR low at high frequency. A DFE performs non-linear operation and subtracts the tail of the signal on the time domain. The subtraction itself does not amplify noise, but as shown in the MFB derivation part of Chapter I, the residual eliminated by DFE can also serve as part of the signal power. Thus DFE also degrades the SNR at receiver.

The development of the phase equalization scheme can be best explained in the frequency domain for linear analysis. As seen in Fig. 3.2, when a noisy waveform is provided as input to a linear receiver filter, both the useful signal and noise react to the filter transfer function. The effect on the signal and inter-symbol interference (ISI) can be seen in the form of signal transfer function (STF) and noise transfer function (NTF). Such analysis is very commonly seen in the design of delta-sigma modulators, *Pavan et al.* (2016), and phase-lock loops (PLL) *Gardner* (2005).



(a) Operation of CTLE on Frequency Domain



(b) Operation of DFE on Time Domain

Figure 3.1: Operations of linear filter based equalizers

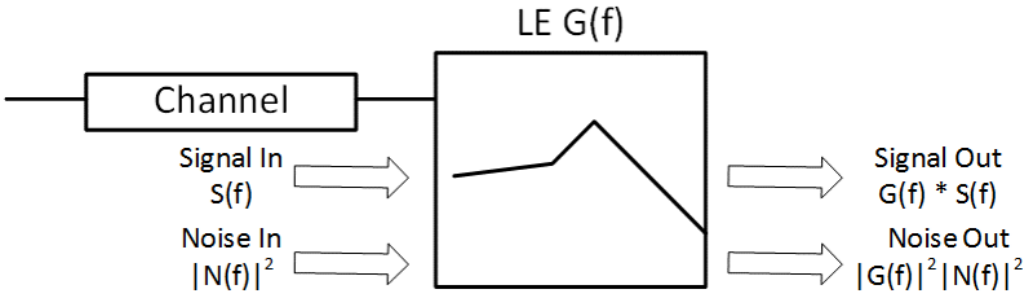


Figure 3.2: Transfer functions of a linear filter on signal and noise

The key observation is that STF and NTF only differ in phase. Therefore, if an equalization scheme affects only the phase while reducing the ISI, the equalization will minimally degrade SNR. In this chapter we discuss a phase equalization scheme along this line.

3.3 Design of A Phase Equalizer

The design of a simple phase equalization can be started with an RC-CR network, a structure often seen in Hilbert filters used in wireless applications *Razavi* (2012). As shown in Figure 3.3a, an RC-CR network gives two waveform outputs. One is the high-frequency path, which is similar to the output of a differentiator, and the other one that is further low passed and thus spreads further. Figure 3.3b shows the simulation result with a set of wireline channel s-parameters. An important observation is that the high-frequency component leads the low frequency-component. This is because it can be shown that when the resistors and capacitors are well matched, the high-frequency component is proportional to the derivative of the low-frequency component.

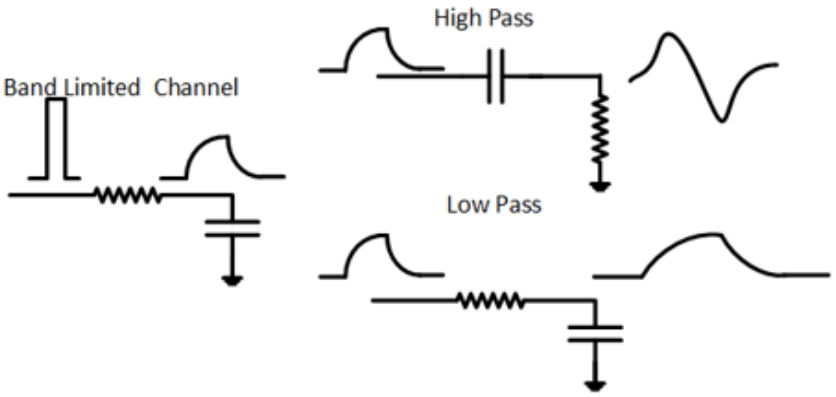
As also shown in Figure 3.3b, if one delays the high-frequency path and recombine with the low-frequency path, so that the peak of the two coincide, about 50% higher peak value and 30% less residual ISI can be observed. The waveform also appears to be more symmetric, showing that phase correction has taken effect.

3.4 System Model of the Phase Equalizer

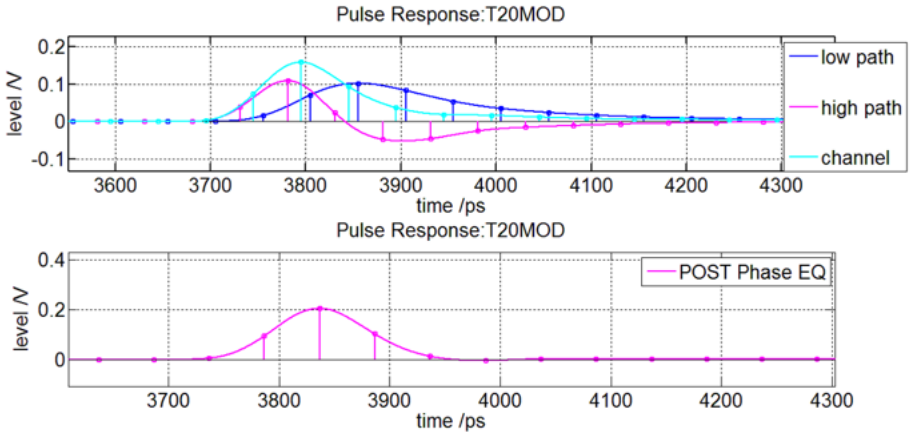
The transfer function of the phase-shaping equalizer can be shown to be

$$\mid H(\omega) \mid^2 = 1 + \frac{2\omega RC sin(\omega T_d)}{1 + \omega^2 R^2 C^2} \tag{3.1}$$

Where T_d is the amount of delay applied to the high-frequency path. Thus the



(a) An RC-CR network in tandem with a low pass channel



(b) Time Domain Simulation of RCCR response to a 20dB loss server backplane channel

Figure 3.3: Signal response to an RC-CR network

transfer function is all-pass at high and low frequency or if T_d is small enough. Therefore this equalization scheme suppresses ISI with almost no loss of SNR.

3.5 Behavioral Simulation

Fig. 3.3 shows the behavioral simulation of the proposed phase equalization scheme on a S-parameter model extracted from real measurements. The channel shows an insertion loss of 26dB at 8GHz and has been used for simulating 16Gb/s NRZ signaling schemes. In Fig. 3.3b, the response of the two paths of the RC-CR has been shown to scale and aligned with realistic phase. Fig. 3.3b also shows the

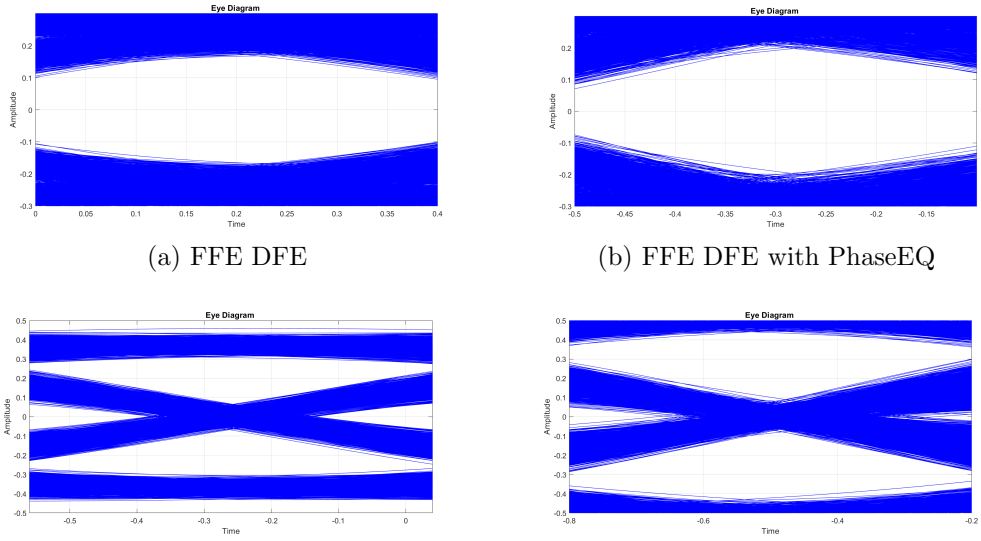
results of delaying the high frequency (HF) waveform and combining the two paths with aligned peak value. A peak distortion analysis shows that 30% boost on the main cursor and 50% mitigation on the trailing ISI's have been achieved through this operation. Note that lowering the trailing ISI's does not only save the currents on the DFE summer,but also helps the linearity requirement on the summing and comparator stages due to the less need for ISI cancellation.

Fig.3.4a and 3.4b shows the eye diagram with a 5-tap FFE and a 7-tap DFE, with PRBS-31 stimuli and taps adapted with least mean square (LMS) iterations. We can observe about 10% extra eye opening on top of the solution without phase equalization. The additional eye opening comes without the cost of noise amplification or error propagation as in CTLE and DFE, and could be helpful when the channel is already sufficiently equalized on the magnitude.

Fig.3.4c, and 3.4c shows the eye diagram comparison under duo-binary detection, where an eye-opening is created based on three levels instead of two. Detecting data under a low-pass signaling assumption, duo-binary schemes are believed to be an SNR and power efficient signaling and receiving scheme, and are often used to implement energy efficient wireline tranceivers, i.e. where power budget on equalization is limited. From the comparison, we observe that the proposed phase equalization scheme is especially beneficial to duo-binary signaling, due to the recycling of energy from the tail to the main tap.

3.6 Implementation of Phase Equalizer

The proposed implementation scheme of phase equalization is shown in Fig. 3.5. The design can be derived from two-path CTLE *Liu and Lin* (2004a), where low-frequency component of the receiver input is subtracted from the main path to obtain high path transfer functions. Traditionally, two-path CTLE is used to decouple the DC gain from the output pole as opposed to the case in source degenerated CTLE



(a) FFE DFE

(b) FFE DFE with PhaseEQ

(c) Duobinary Eye for FFE DFE

(d) Duobinary Eye for FFE DFE and PhaseEQ

Figure 3.4: Simulated Eye Diagram for a 26dB loss channel with Different Equalizer Setup.

Gondi and Razavi (2007). In our design, given a two-path CTLE, we can observe that both the low frequency (LF) and high frequency (HF) signals are obtained in one circuit, and only one addition, a delay element is needed to delay the HF by a desired amount.

The two-path CTLE approach is superior to a baseline RC-CR direct implementa-

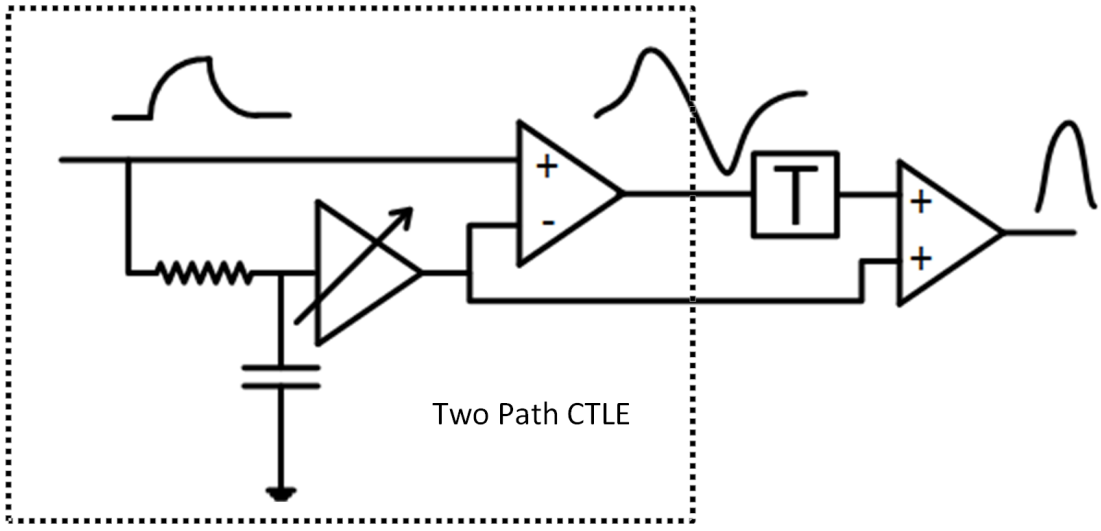


Figure 3.5: proposed implementation of phase equalization

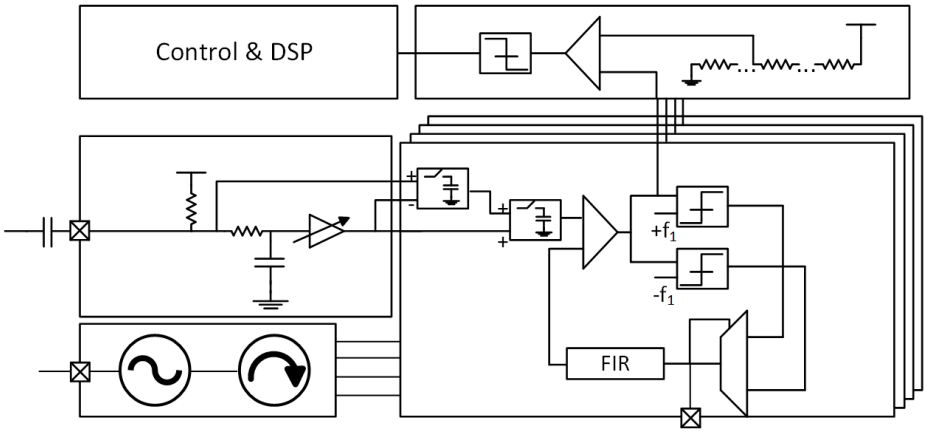


Figure 3.6: RX Implementation Top Level

tion. The direct implementation of RC-CR network suffers from parasitic mismatch and therefore would require significant, i.e. near 40% tuning to ensure proper transfer function implementation. Implementations based on analog delay element *Boesch et al.* (2016), using opamp based bi-quad structures, can also be used and could potentially provide more flexibility of transfer function design. But this approach is power hungry and could easily double the power consumption of the overall system.

One further simplification is possible when half or quarter rate receiver architecture is implemented. The delay element can be conveniently replaced by a sample-hold circuit (SH) to delay the HF by 1/2 or 1/4 of UI. Since most of high speed designs are indeed at least half rate, this design is readily applicable to reduce design complexity and power on clock distribution. The power saving can be even more significant when multiple lanes on a chip share a common clock source.

3.7 Implementation of a Phase Equalization based receiver

As a proof of concept, a complete phase EQ based receiver has been implemented. The receiver uses a quarter-rate architecture, with a phase EQ front-end and a 7-tap quarter rate DFE. Fig. 3.6 shows the circuit implementation of the phase EQ, including a charge-steering SH *Bai et al.* (2014) and two input buffers.

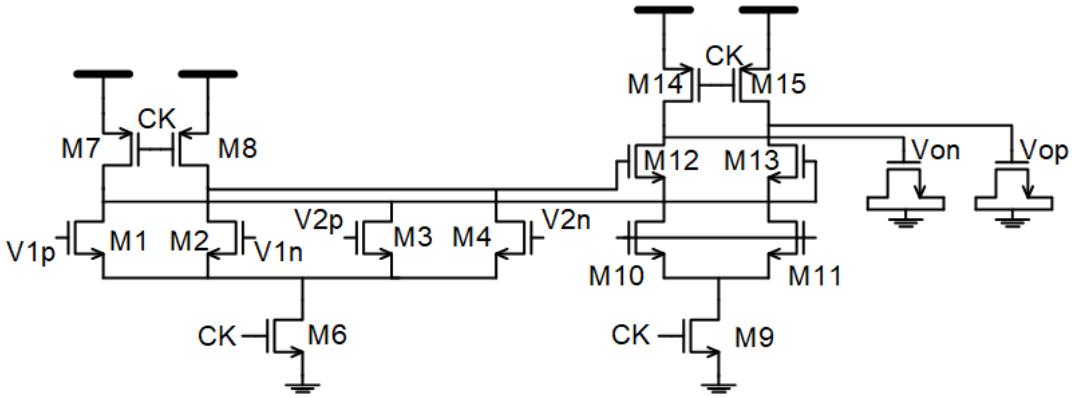


Figure 3.7: Implementation of SH and Sampler

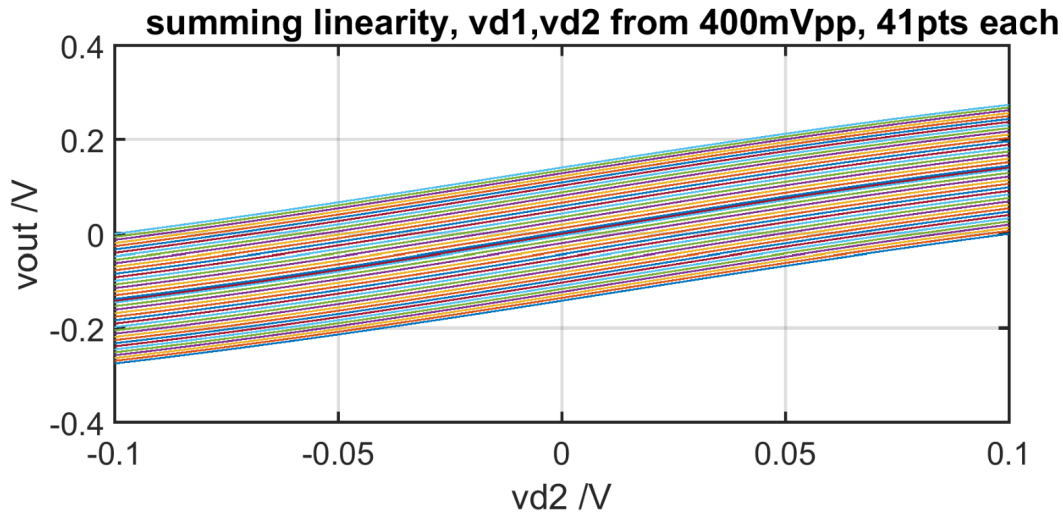


Figure 3.8: Input/Output Relationship of the Summing SH

Fig. 3.7 shows the design of the charge steering sample and hold circuit (SH). In our implementation the SH is modified to take two inputs and thus performs sampling and addition at the same time. Figure. 3.8 shows the input output characteristics of the proposed SH and it shows that the operation has good linearity with respect to both inputs at a 400mVpp input range.

The sampling function (SF) of the SHs are shown in Fig. 3.9. The SF characterizes the SH circuit's response to the change of input signal over the spectrum. The difference between the SF and the settling time of the circuit is that the SF characterizes the signal quality while the settling time only shows how fast the circuit can

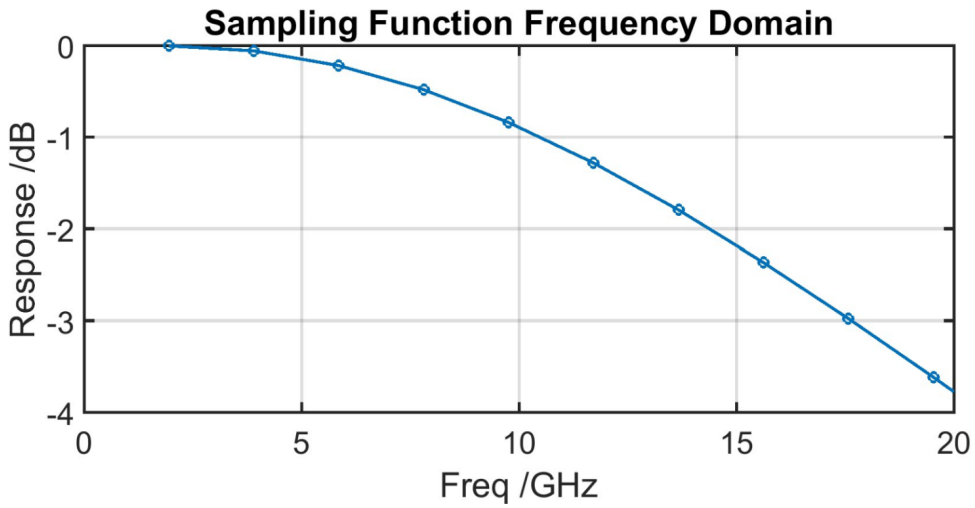


Figure 3.9: Sampling Function of the SH

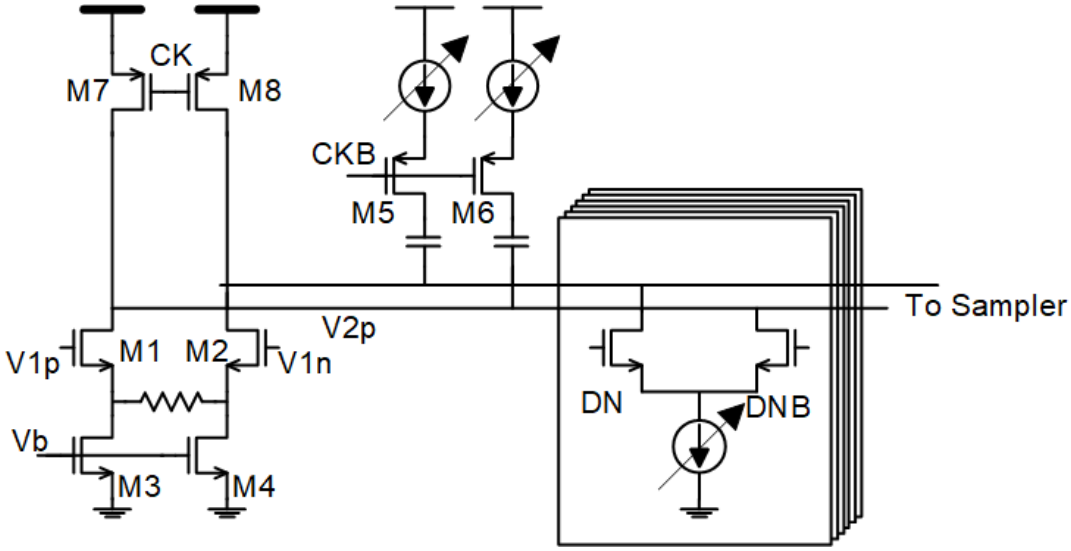


Figure 3.10: Implementation of DFE

operate. In designing communication related circuits, the SFs are to be kept ideally just above necessary to minimize power dissipation and noise amplification.

Fig. 3.10 shows the implementation of the DFE. The DFE operates by integrating the input signal and runs in two alternating phases. In the reset phase, the output is held high by the clock switch. In the integrating phase the output signal slowly integrates the input for one UI and the common mode of the output is kept high enough for the next stage by a common mode restoration circuit. The integrating

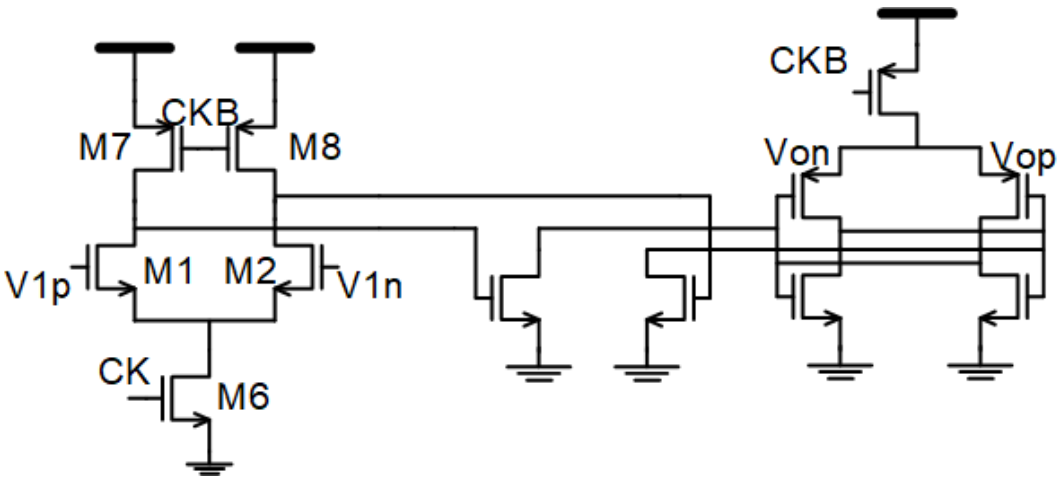


Figure 3.11: Implementation of Comparator

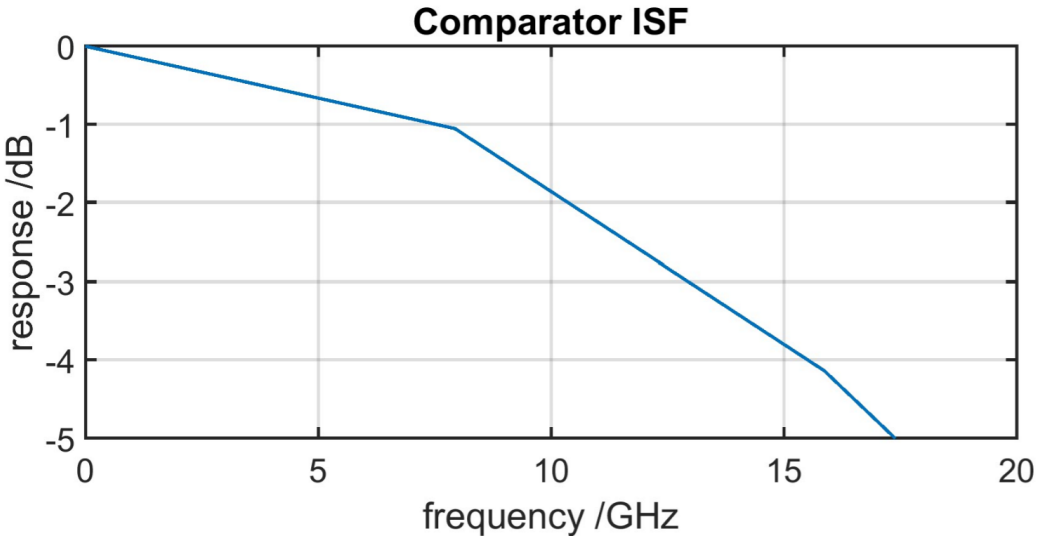


Figure 3.12: Impulse Sensitivity Function of the Comparator

DFE has the advantage of reduced bandwidth requirement and thus is popular in energy efficient equalizer designs.

Fig. 3.11 shows the sampler implementation. A double-tail latch design has been chosen to achieve high gain and thus high speed. Figure. 3.12 shows the impulse sensitivity function (ISF). An ISF is similar to the sampling function (SF), and in designing samplers, it is also ideal to keep the ISF band-limited while providing enough sensitivity to the signal input. A low bandwidth ISF indicates that the sampler is going to introduce more ISI while an excessively high ISF bandwidth makes the com-

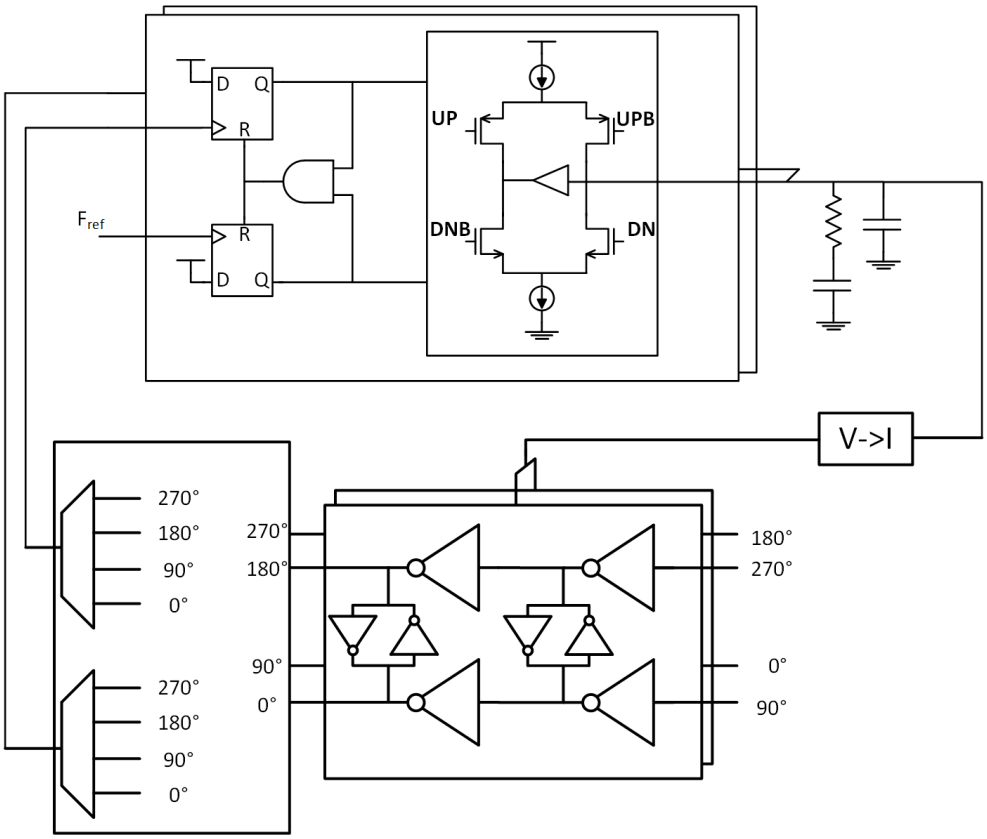


Figure 3.13: PLL Top Level

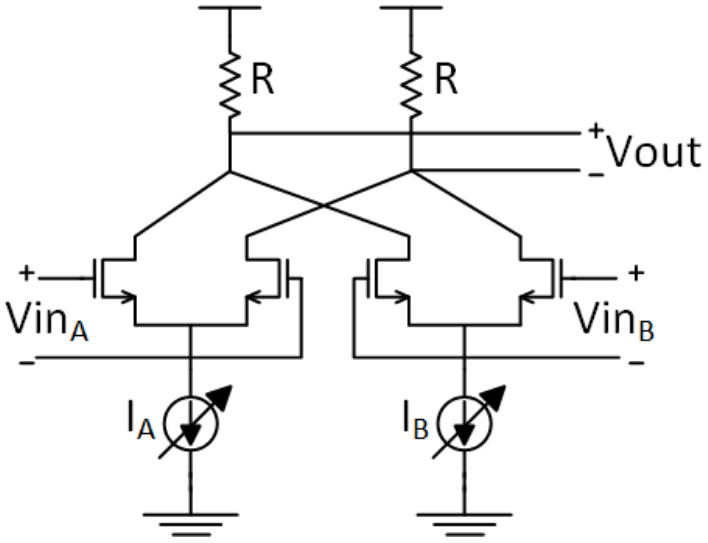


Figure 3.14: Conventional Phase Interpolator Design

parator more sensitive to noise and interference such as signal coupling from other paths, and supply noise.

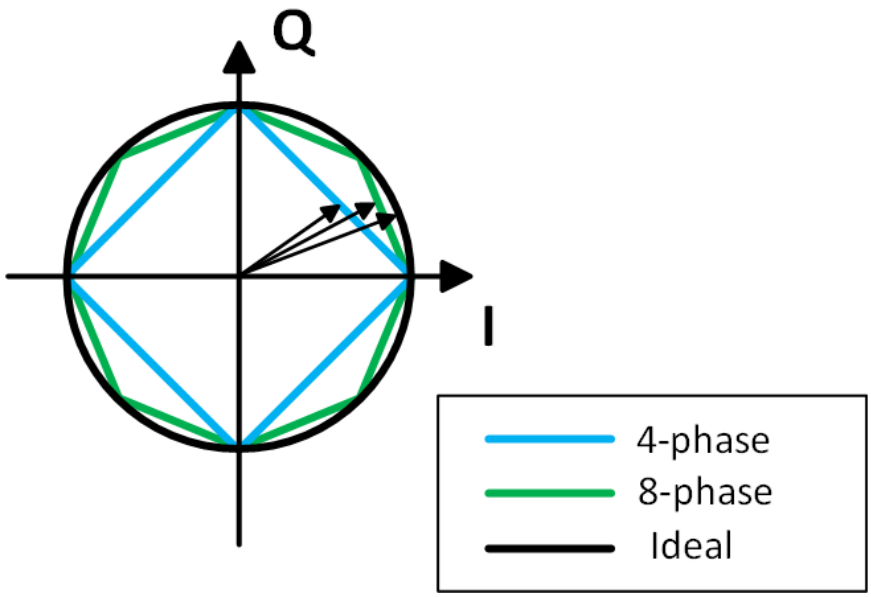


Figure 3.15: Characteristics of ideal and approximate phase interpolation

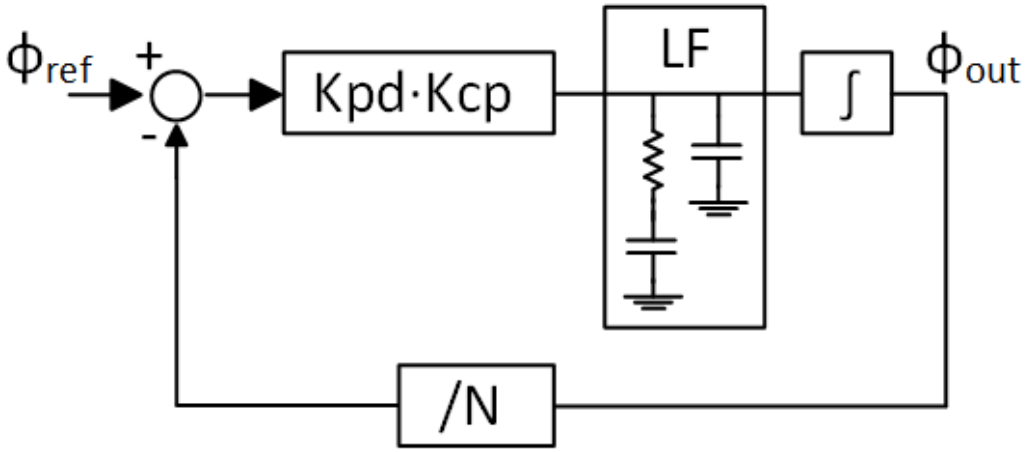


Figure 3.16: phase domain model of a PLL

Fig. 3.13 shows the design of on-chip PLL for generation of 4 phase clock while keeping synchronization to the TX. A quarter rate forward clock input is assumed and a dual-PFD loop design *Toifl et al.* (2005) is adopted to achieve 8bit true linear phase interpolation.

Phase interpolation (PI) plays an important role in the clock data recovery (CDR) in a wireline communication system. The output of the PI controls the sampling

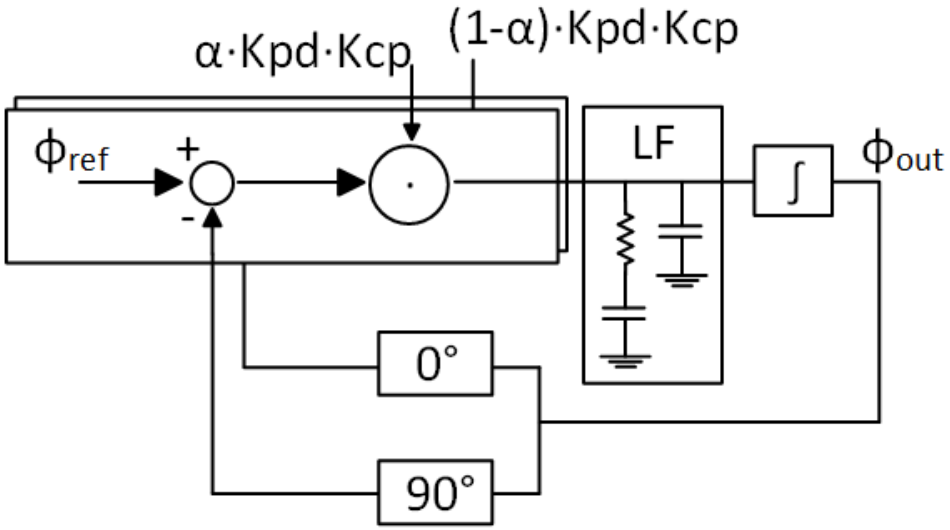


Figure 3.17: phase domain model of a PLL

phase of the discrete time circuits and thus needs to be accurate and low noise. As the requirement on timing systems become higher in each generation of the wireline communication products, more precision is needed to support advanced CDR algorithms. Today's standards typically require 7b fine grain interpolation within one unit interval, or 50ps for a 20Gb/s design.

Figure. 3.14 shows a commonly used circuit for phase interpolation. The circuit essentially linearly combines two signals of different phases and can be modeled by equation.3.2

$$V_{out}(t) \propto I_A \times V_{inA}(t) + I_B \times V_{inB}(t) \tag{3.2}$$

The circuit performs ideal phase interpolation when both of the inputs and sinusoids with the same amplitude and frequency but different phase. For example, suppose

$$V_{inA}(t) = cos(t) \tag{3.3}$$

$$V_{inB}(t) = sin(t) \tag{3.4}$$

hold, an intermediate phase of $V_{out}(t) = cos(t + 10^{\circ})$ is needed, one can simply set

$$I_A \propto cos(10^{\circ}) \tag{3.5}$$

$$I_B \propto sin(-10^{\circ}) \tag{3.6}$$

However, this approach requires complex trigonometric computation. To simplify the task, typical designs use approximate interpolation with the parameter set as follows:

$$I_A = \alpha \times I_{bias} \tag{3.7}$$

$$I_B = (1 - \alpha) \times I_{bias} \tag{3.8}$$

$$0 \leq \alpha \leq 1 \tag{3.9}$$

which essentially approximate phase interpolation by linear waveform interpolation. The accuracy can be improved by choosing the two inputs to be closer in phase, i.e. 45° apart instead of 90° apart in the example case. Figure. 3.15 compares the characteristics of ideal interpolation and approximate interpolation with finer phase spacing.

In our design we implement a phase locked loop (PLL) based phase interpolation. The system model of the design is shown in Figure. 3.16. A PLL is a second order control loop with respect to the phase of the output signal. When in lock condition, the loop forces Equation.3.10 to hold.

$$\phi_{ref} - \phi_{out}/N = 0 \tag{3.10}$$

This characteristic is ideal for adjusting the phase of the oscillator output because the second order behavior forces the phase error to be asymptotically zero.

To implement phase interpolation in the PLL, we introduce two paths for phase comparison *Toifl et al.* (2005). As shown in Figure. 3.17 each of the phase comparison paths compares the reference clock with two phases from the oscillator and the lock condition becomes:

$$(\alpha)(\phi_{ref} - (\phi_{out} + 0^\circ)) + (1 - \alpha)(\phi_{ref} - (\phi_{out} + 90^\circ)) = 0 \tag{3.11}$$

and it simplifies to:

$$\phi_{ref} - \phi_{out} = \alpha \times 0^\circ + (1 - \alpha) \times 90^\circ \tag{3.12}$$

This approach directly interpolates phase and therefore does not suffer from approximation error. Moreover, to accommodate for the 4-phase interleaved design of the receiver path, the clocking path needs to provide 4 phases of the clock. This approach enables the four phases to be tuned simultaneously while maintaining their 90° difference relationship.

3.8 Simulation Results and Comparison

Fig.3.18 shows the layout of the RX design and table and Fig. 3.19 shows the power breakdown of the RX system running at 16Gb/s operation. The overall power consumption of the system is 16.14mW which translates to a 1pJ/bit energy efficiency, which is competitive against the state-of-the-art design *Bai et al.* (2014). Also the proposed design did not explore the use of voltage scaling as did in *Bai et al.* (2014) and more taps were included in the DFE to provide stronger equalization. The largest part of the power consumption is due to clock generation and distribution, which can be further reduced with optimized clock buffer designs and also low jitter PLL archi-

tectures. The data path power consumption is dominated by the comparator circuit, whose power consumption depends on the system requirement, i.e. the linearity, sensitivity and mismatch. The implemented comparator was slightly over designed to cover nearly 10mV eye opening with less than 5mV mismatch, For lower loss channels with 10 to 15dB of insertion loss, significant power can be saved due to reduced sensitivity and mismatch.

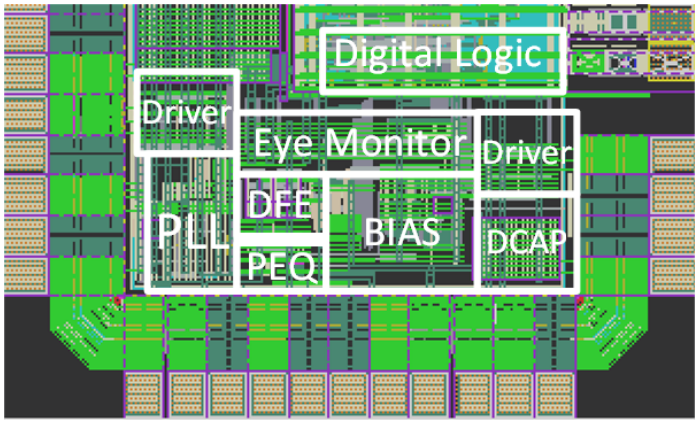


Figure 3.18: RX Top Level Layout

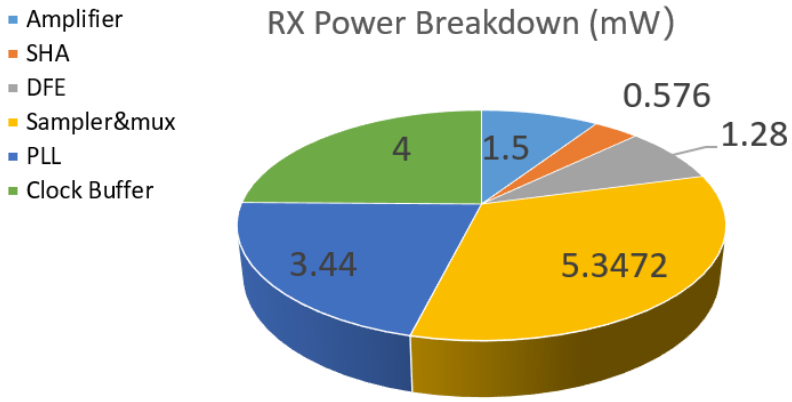


Figure 3.19: RX Power Breakdown

3.9 Conclusion

In conclusion, a novel phase domain equalization concept is proposed to mitigate ISI without boosting high frequency noise. The performance of the equalization

scheme has been characterized in behavioral simulations. Circuit implementation of the proposed equalization scheme has been provided and a complete RX solution based on the proposed concept has been implemented up to layout and characterized.

CHAPTER IV

LEIA:Parallel Lattice Encryption Instruction Accelerator

4.1 Introduction

Almost all public key cryptosystems today rely on the hardness of integer factorization and discrete logarithm, both of which are known to be efficiently solved by Shors algorithm performed on a quantum computer *de Clercq, et al.* (2015). With quantum computers being closer to the public access than ever, a leap into post-quantum security is not only timely but also necessary(Fig. 4.1).

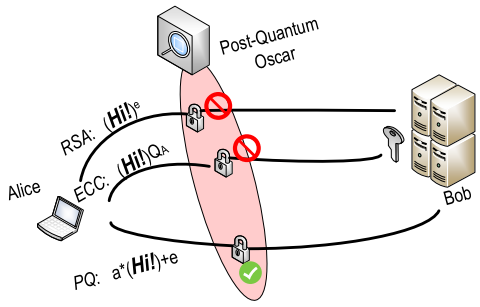


Figure 4.1: Threat of Quantum Attacks

The emerging lattice-based cryptography is widely believed to be a quantum-resilient alternative to classical public key cryptosystems *Verbauwhede, et al.* (2015).

Ring learning with errors (ring-LWE) is the most-recognized lattice-based cryptographic scheme and it holds promise towards practical fully homomorphic encryption *Liu, et al.* (2017). In Figure. 4.2, the procedure of RLWE key exchange is illustrated. When two users or hosts, Alice and Bob, need to establish a shared key through a public channel, they each simply multiply the public key with their own private key and then add some small noise vector. Upon receiving the transmitted vector from the channel, they each multiply the received vector again with their own private key and they thus arrive at an agreement up to some small errors.

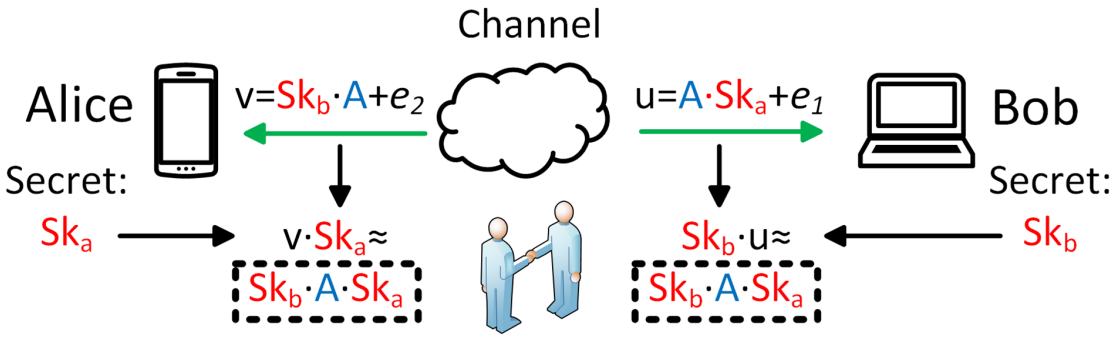


Figure 4.2: Ring Learning with Error (RLWE) key exchange

Meanwhile suppose a third-party user of the channel, Oscar, has been collecting all the information distributed by the channel, and wants to arrive at the same shared key (Figure. 4.3). This task involves solving the ring learning with errors (RLWE) problem, and under certain constraint on the public key and the noise vectors, RLWE can be extremely difficult. Thus the security of RLWE based key exchange (KEX) is provided.

In this work, we present LEIA, the first ring-LWE instruction processor IC to support a common set of ring-LWE based security applications, including digital signature and key exchange, consuming 140mW, providing up to 200x acceleration and 50x more energy efficiency over state-of-the-art implementations on processors, FPGAs, and embedded devices *Alkim, et al.* (2016); *de Clercq, et al.* (2015); *Liu, et*

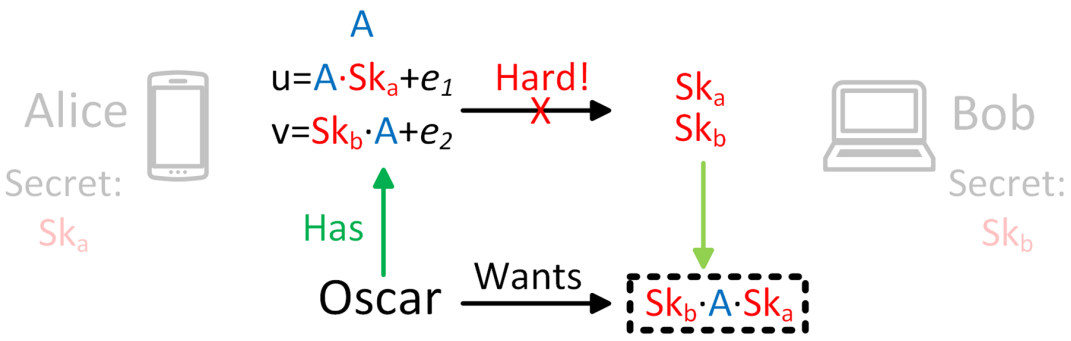


Figure 4.3: Reversing RLWE key exchange

al. (2017); *Oder, et al.* (2014); *Roy, et al.* (2014); *Verbauwhede, et al.* (2015).

Ring-LWE based cryptosystems are built using polynomial operations on cyclotomic rings of order N on Rq, where q is a prime number and N is assumed to be of power of 2. To secure a message, cyclotomic ring polynomial multiplication is used to rotate the message, and high-precision discrete Gaussian (DG) samples are added to mask the message. When N is a power of 2, cyclotomic ring polynomial multiplication can be done in O(NlogN) with number theoretic transform (NTT). NTT is compute-intense and is often replaced using floating-point FFT libraries, which is prone to numerical underflow and clipping issues. High-precision DG generation is slow and requires a large memory. A survey of a recent implementation of a ring-LWE based cryptosystem (Fig. 4.4) reveals that NTT and DG generation represent 43% and 39% of the workload, respectively *Alkim, et al.* (2016).

4.2 Proposed Solution

4.2.1 LEIA Overview

LEIA(Fig. 4.5) is designed to overcome the challenges by providing a 16x parallel, up to 32 points/cycle, configurable NTT accelerator supporting N up to 2048 and q up to 2^{32}, and a 4x parallel DG accelerator with an average 2-cycle DG sample generation

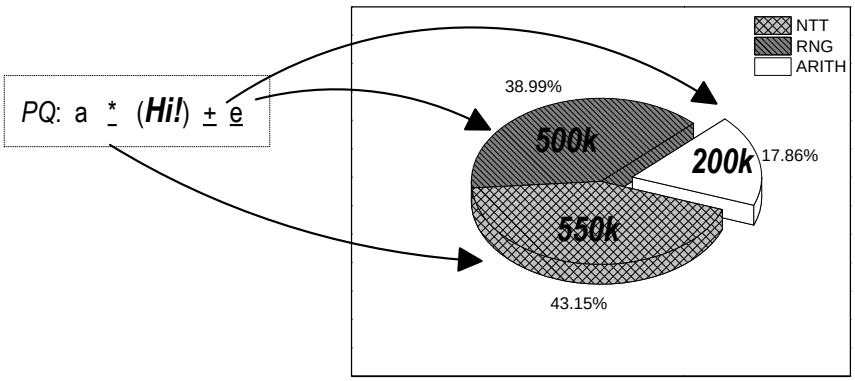


Figure 4.4: Cycle Count Breakdown of NewHope*Alkim, et al.* (2016) on ARM Cortex-M4

and configurable accuracy up to 256 bits for near 256-bit secure cryptosystems. By iterating between NTT and DG, LEIA supports distributions of extended standard deviation by another 10x using Micciancios precision-extension scheme*Micciancio, et al.* (2017). LEIA also contains a hash-function core, a central controller and a main memory. LEIA is programmed by instructions to support a variety of security applications through configurable blocks and dataflow.

4.2.2 The NTT Core

The integer polynomial multiplication task is achieved through a number theoretical transform (NTT) aided approach (Figure. 4.6). Direct computation of the coefficients involves circular discrete convolution which has a complexity of $O(N^2)$. With the help of NTT, the overall complexity can be reduced to O(logN), as illustrated in Figure. 4.6.

The NTT computation resembles FFT, and it is done using stages of integer butterfly operations. We choose decimation-in-time (DIT) for forward NTT and decimation-in-frequency (DIF) for inverse NTT, both of which reuse the same hardware. Our NTT architecture is constructed in hierarchical stages(Fig. 4.7a) using an

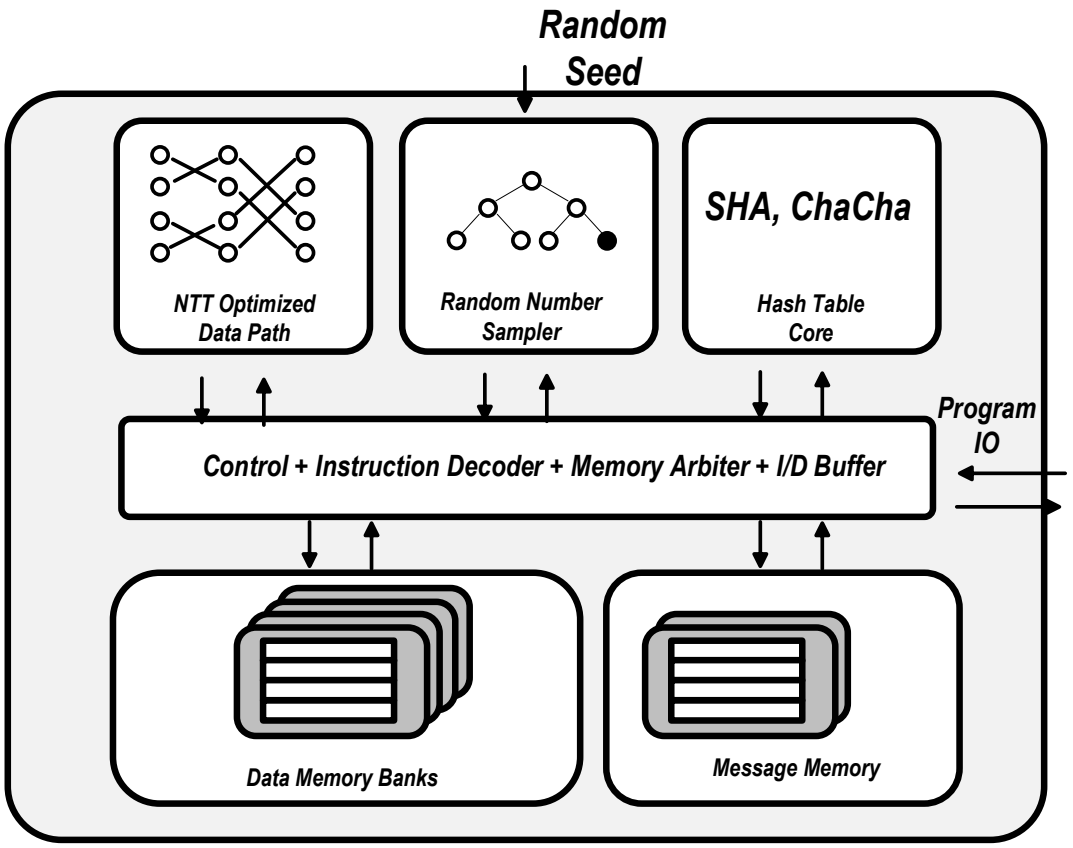


Figure 4.5: Proposed LEIA Architecture Top Level

array of simple NTT PEs(Fig. 4.7b). Each NTT PE is 2x parallel and performs two butterfly operations using two Montgomery multipliers, four q-ary integer arithmetic units for modulo-q adds or subtracts, and a 4-port local scratch pad to provide efficient butterfly connections. The size of the scratch pad determines the size of NTT that a PE can handle locally. To save area, we limit the scratch pad to 64-entries to support up to 64-point NTT locally. Beyond 64-point NTT, we use multiple NTT PEs and short routers to exchange data between PEs. However, longer NTTs result in longer wiring and routing becomes increasingly inefficient. Therefore, we limit to only 8 NTT PEs to support up to 256-point NTT. Above 256-point inputs (power-of-2) are divided into vectors of 256, and the 8 NTT PEs are used as a 256-element vector processor. In summary, the NTT computation is divided to three stages to support

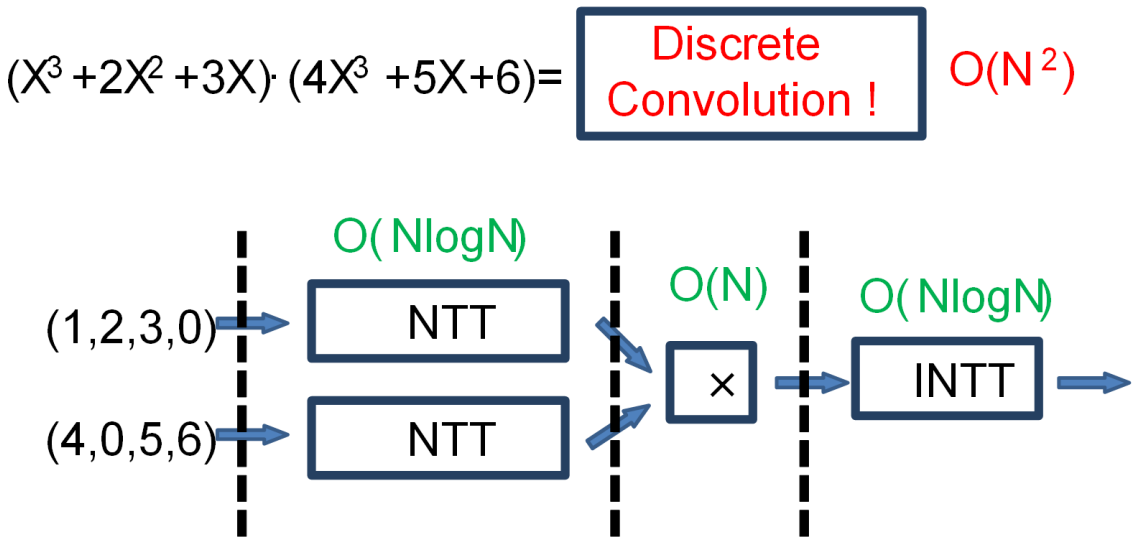
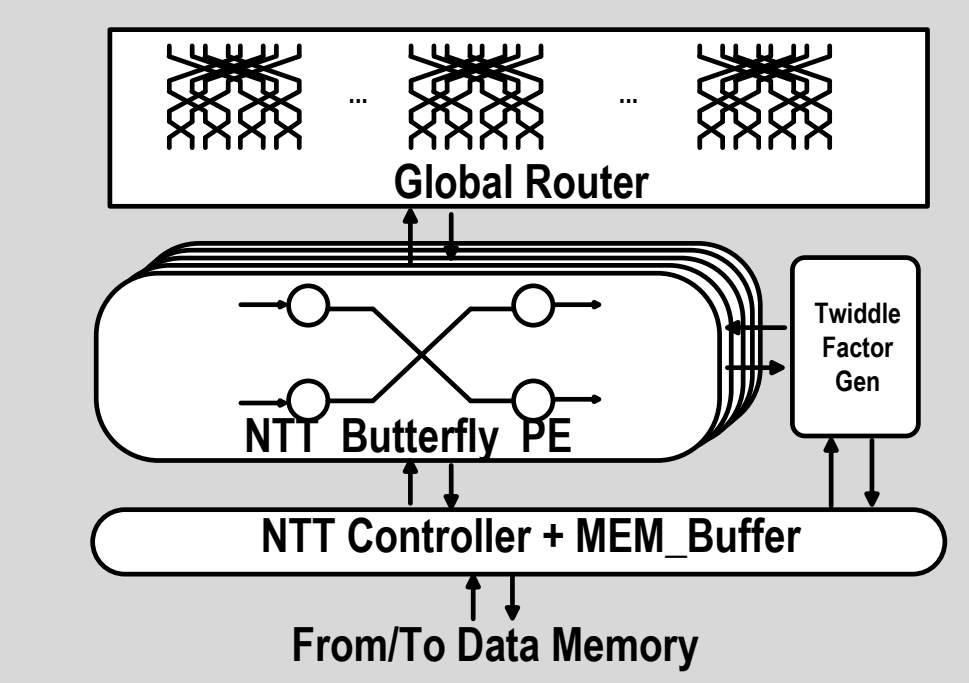


Figure 4.6: Integer polynomial multiplication through NTT

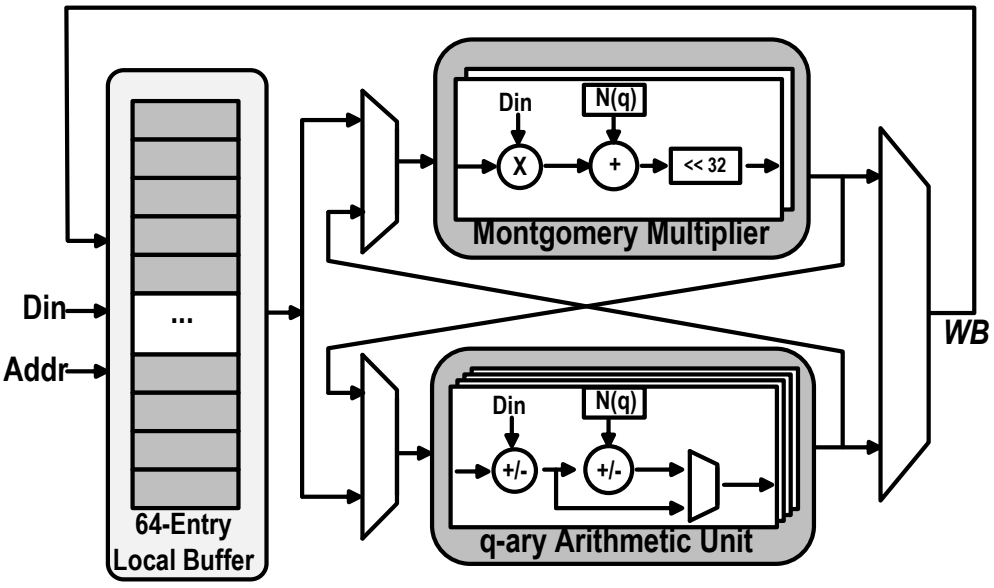
efficient reconfiguration, as illustrated in Fig. 4.8a: short NTT up to 64-point using only local processing, intermediate NTT up to 256-point using short routers, and long NTT up to 2048-point using vector processing. The three-stage architecture balances local storage and global routing, as well as data routing and processing delay, resulting in nearly optimal tradeoff between latency and parallelism as shown in Fig. 4.8b. Doubling the parallelism to 32x (i.e., 16 NTT PEs) improves latency of a 256-point NTT by only 37%, showing fast diminishing returns. The NTT-optimized data path is generic and supports all the instructions needed to fulfill lattice encryption tasks (Table. 4.1)

Table 4.1: Data Path Instructions

Instruction	**Cycle Delay**	**Functionality**
REG_RD	16	Load MEM Data
REG_ST	16	Write Data to MEM
ADDSUB	20	Modulo Vector ADD/SUB
NTT	160	256-Point NTT
INTT	200	256-Point INTT
MULT	20	Modulo Vector Multiplication
DGREAD	20	Load DDG Data



(a)



(b)

Figure 4.7: (a) Proposed NTT-Optimized Data Path, (b) Proposed NTT-Optimized Arithmetic PE

4.2.3 The DDG Tree Implementation

Sampling from discrete Gaussian (DG) distribution with very high precision can be a difficult task. In this context, the precision requirement is understood to be

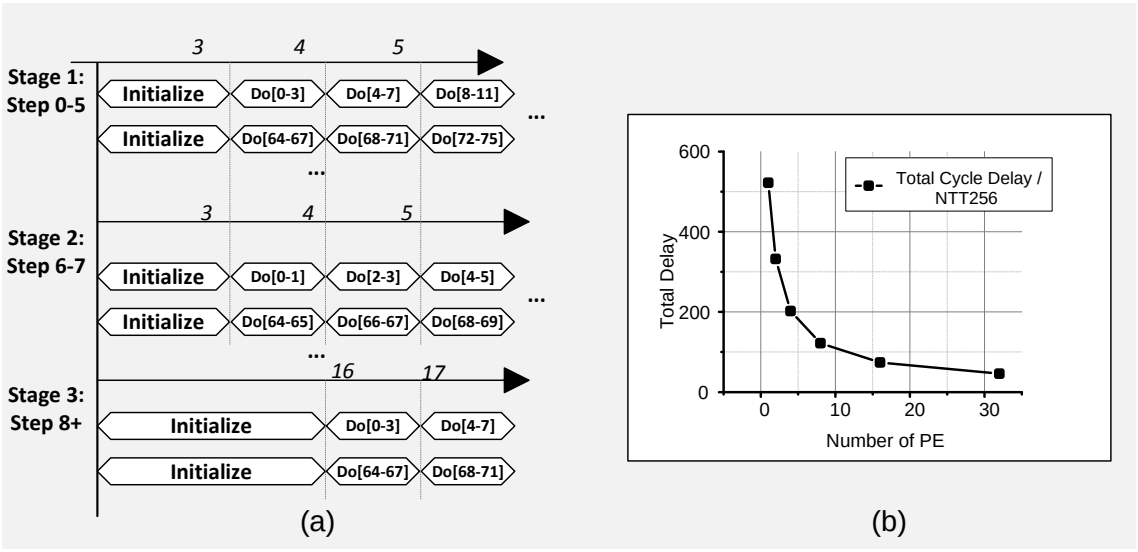


Figure 4.8: (a) Timing of Each PE at Each NTT Stage (b) Cycle Delay of NTT OP versus Number of PEs in the Core

the statistical distance between the actually realized distribution and the required theoretical random variable, as in Def. IV.1.

Definition IV.1 (Statistical Difference)**.** Given a random variable X with probability density function (PDF), $P_X(x)$, with the sample space Ω, a realization $\hat{X}$ can be obtained through a practical sampling process. And the **statistical distance** between X and $\hat{X}$ is defined as

$$\mathbf{SD}(\mathbf{X}, \hat{\mathbf{X}})_i = \mathbb{E}_{x \in \Omega}(|P_X(x) - P_{\hat{X}}(x)|) \tag{4.1}$$

Meanwhile, some applications may call for more stringent version of the requirement on precision and in our design, we design for accommodating the worst case scenario where the expectation operator is replaced by a maximum.

The task of the DG accelerator is to transform a given random bit stream into DG distributed sequences (Figure. 4.9). The design of this module should satisfy the requirement outlined in Def. IV.1. Meanwhile, the throughput and efficiency need to be optimized to ensure both compatibility with most applications and low power

consumption.

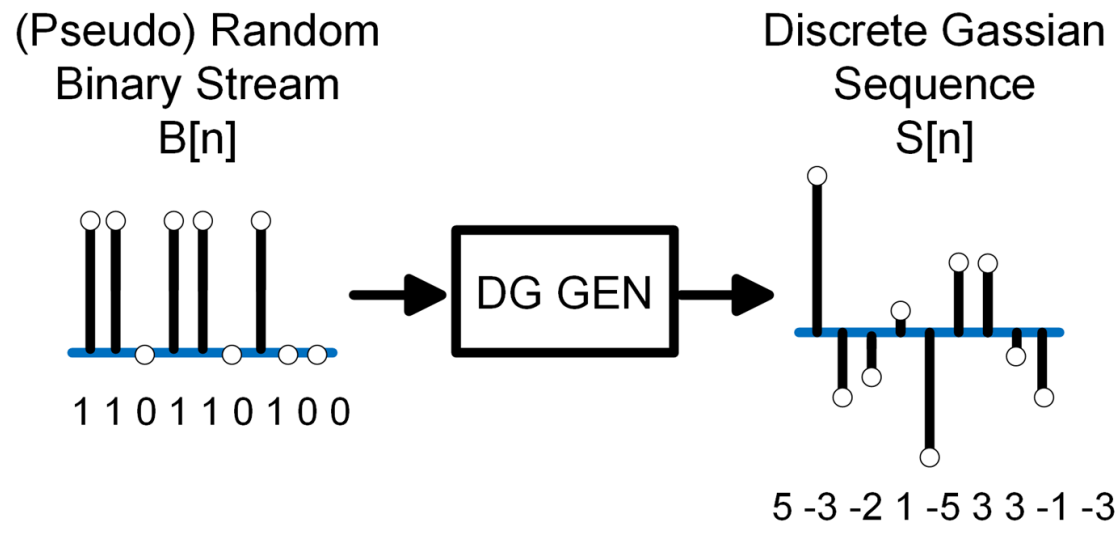


Figure 4.9: DG Sampling Operation

Another metric to consider is the average number of random bits needed to produce one sequence sample. A naive DG implementation is to invert the cumulative mass function (CMF) of the discrete distribution, as shown in Figure 4.10. One block of random bits can be grouped to represent a number from zero to one and a one-step reverse look up yields the wanted sample from DG distribution.

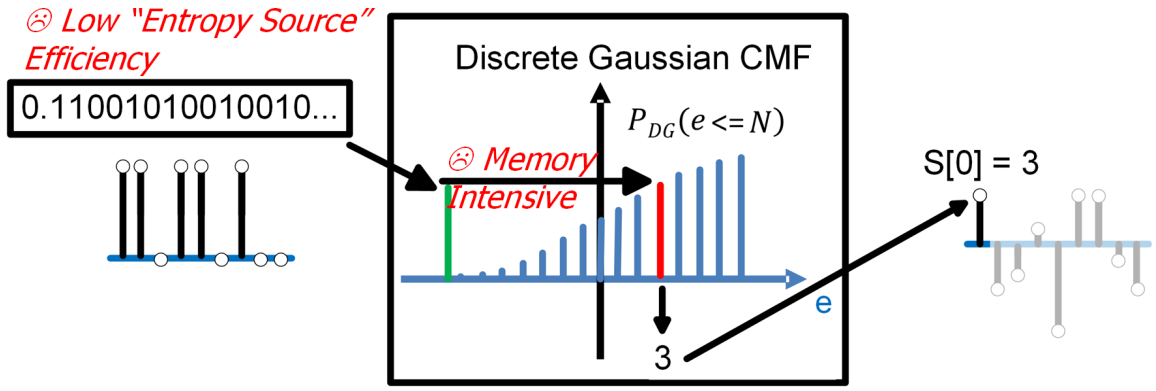


Figure 4.10: Look up table (LUT) based inversion sampling

This method is functionally correct but can be costly in two ways. First, it is not

entropy source efficient, because for an N bit precision, i.e. N can be 80, 128 or 256, N binary bits are needed. However, the absolute minimum number of bits needed for each sample is the entropy of the target distribution, which oftentimes can be much less than N. Second, the table look up complexity scales exponentially with N and can easily become impractical when N becomes large.

The proposed DG accelerator is designed based on Knuth-Yao (KY) Sampling (Fig. 4.11) that is known to be the fastest method for precision random sample generation *Roy, et al.* (2014).

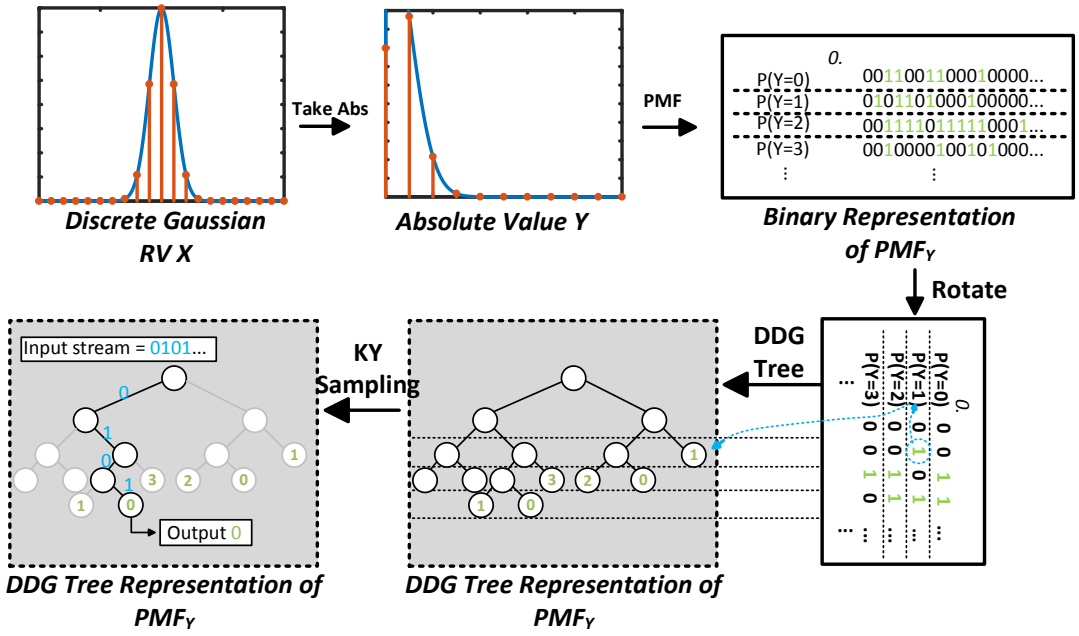


Figure 4.11: Construction of DDG Tree and Operation of KY Sampling

Given a DG distribution, the first step of KY sampling is to represent this distribution as a discrete distribution generation (DDG) tree. A DDG tree is a binary tree with each node representing either a transition or a termination with an output value. The construction of the DDG tree is illustrated in Figure. 4.12. Each layer of the DDG tree corresponds to each layer of the "bit plane" of the binary representation of the PMF of the target distribution. The number of terminal nodes on each layer of

the DDG tree is exactly equal to the number of 1's in each layer of the "bit plane." The construction of the DDG tree can run as deep as needed to satisfy the precision requirement.

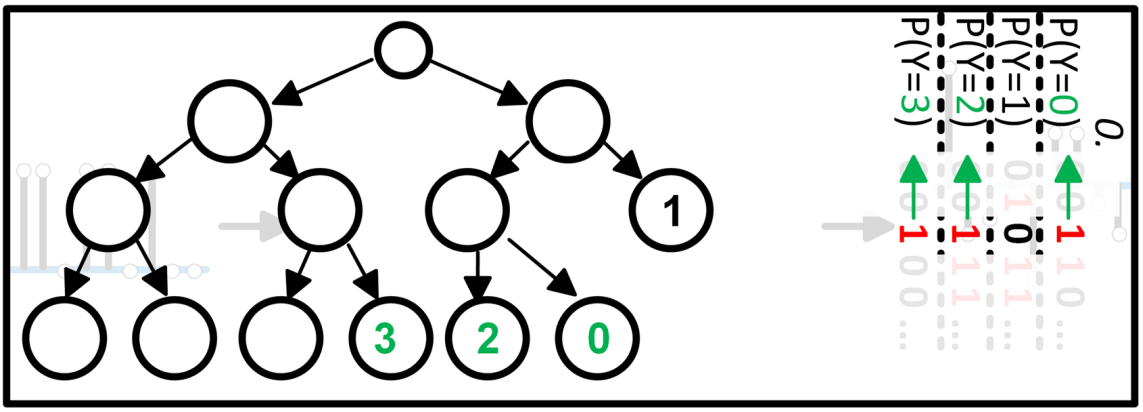


Figure 4.12: Construction of DDG Tree

KY Sampling traverses the DDG tree from the root (Figure. 4.13), taking one binary random bit each time to decide either left or right branch to take next. It outputs a sample of the given distribution when it reaches a non-empty node. The DG accelerator also supports binomial noise generation used in NewHope*Alkim, et al.* (2016).

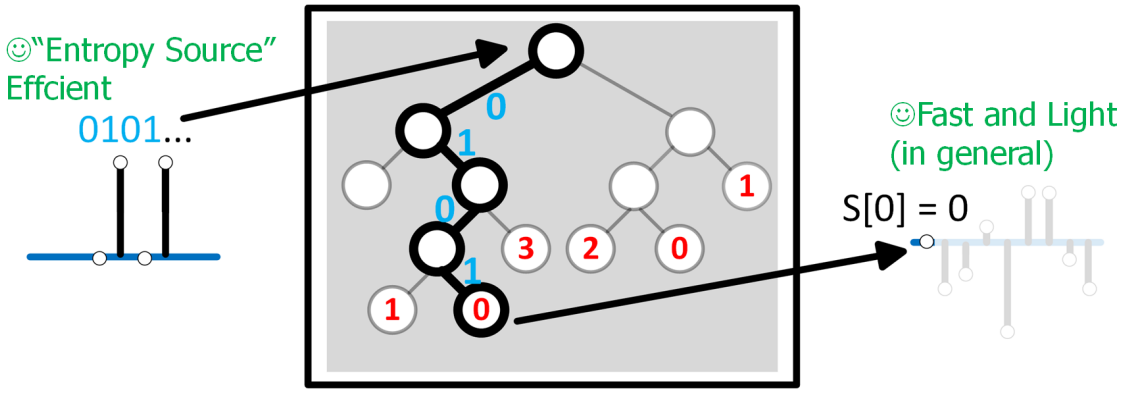


Figure 4.13: KY Sampling using DDG Tree

It can be shown that given an infinitely deep DDG tree, the expected number of

bits used by KY sampling to generate one sample approaches exactly the entropy of the target distribution. Thus, KY sampling is already asymptotically optimal. It can be observed that each node in the DDG tree at layer M has a probability of being visited equal to 2^M (Figure.4.14)

A straightforward implementation of the DG accelerator is costly and slow. Storing the DDG tree for a 256-bit precision DG distribution easily approaches 1Mb of memory, and tree traversal is serial in nature. To enable parallel DG generation, the DDG tree memory needs to be duplicated or additional ports added, both of which incur prohibitive costs in area and power.

We design a fast KY Sampling architecture by taking advantage of a key insight: each level down the DDG tree represents a higher precision representation of the given distribution, and the probability of having to traverse deeper into the tree decreases approximately exponentially.

This leads to a fast and agile KY Sampling architecture consisting of four parallel sampling modules and a 2-level memory hierarchy, as shown in (Fig. 4.15). A sampling module operates two pipeline stages, a tree traversal stage that takes random bits and eventually detects a non-empty node, followed by a value look-up stage to retrieve the value stored in the node. Each module employs a compact, fully searchable local cache

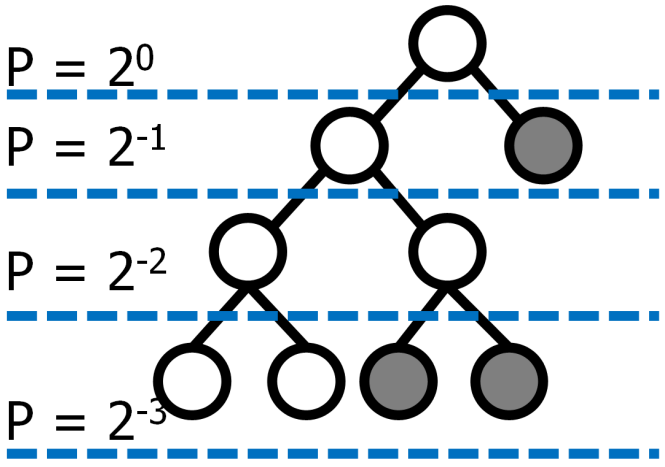


Figure 4.14: Probability of visiting each node on a DDG tree

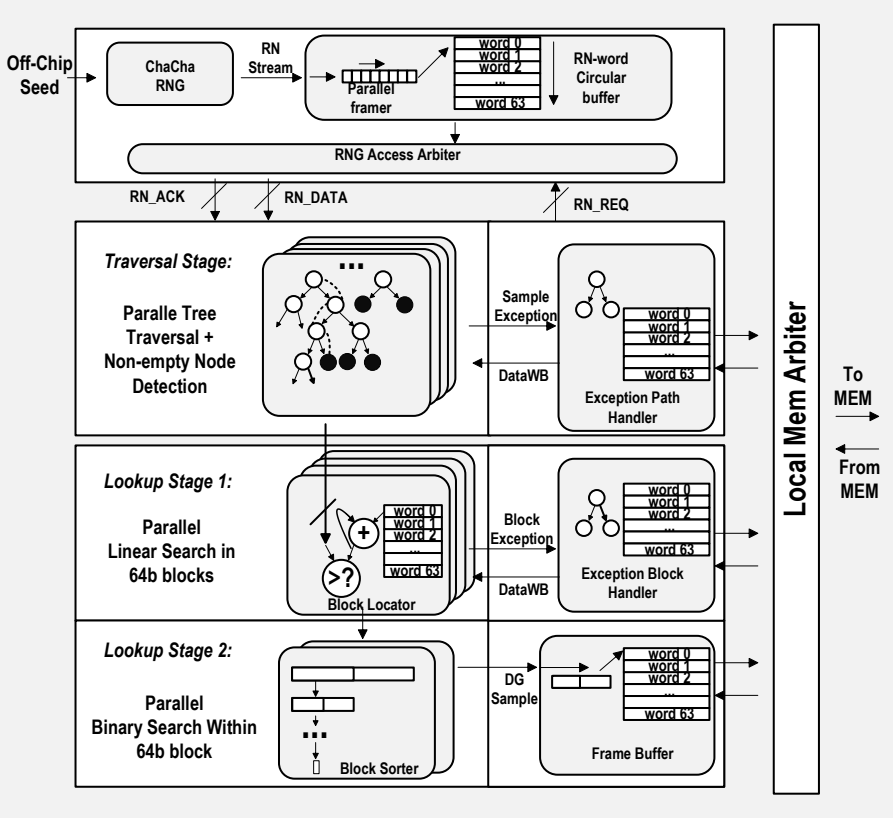


Figure 4.15: Proposed Parallel KY Sampler

that stores only the first 6 levels of the DDG tree to accelerate local tree traversal and minimize storage duplication. Within 1% cache miss rate, a module needs to access the remaining, majority part of the DDG tree that is stored in main memory. An arbiter serializes the highly unlikely case of main memory access contention.

The DDG tree is stored in memory as bit-planes. The Gaussian distribution, stored at high precision, i.e. 256 bits, results in 5% to 6% sparse storage of the DDG tree, which opens door to significant compaction of the bit-plane storage. We create a compression scheme by dividing a bit-plane into 16b vertical slices, and removing all-zero slices. The simple compression scheme saves 93% memory at 256b precision (Figure. 4.16).

For distributions of larger spread that are used in some applications, the first several rows of the bit-plane, i.e., the top levels of the DDG tree, are often entirely

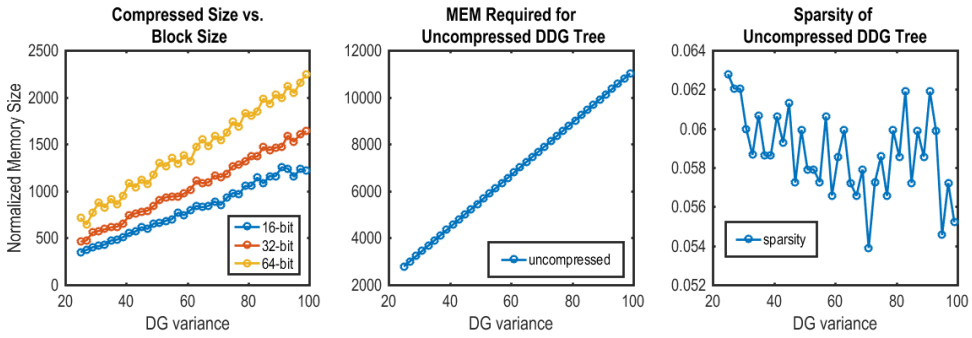


Figure 4.16: Compression of DDG tree

zeros, and thus can be conveniently skipped, maintaining an average 2-cycle DG sample generation latency.

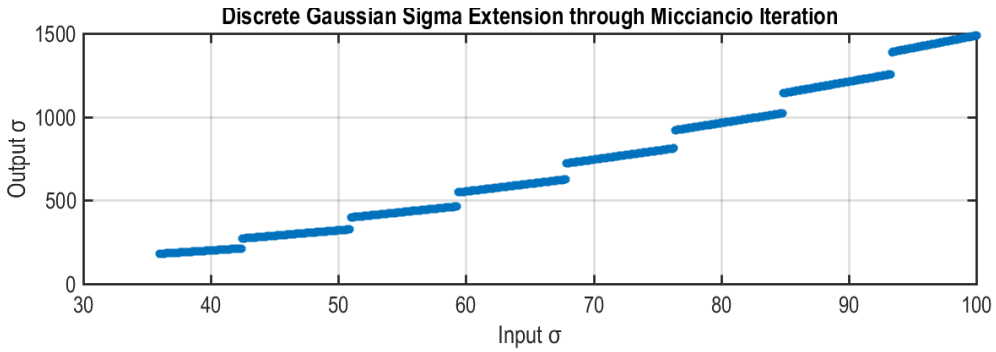


Figure 4.17: Available variance values from current design

A variance extension scheme*Micciancio, et al.* (2017) is adopted in our design to save the cost of generating high variance discrete Gaussian distributions. With the overhead of only two multiplications and one addition, the extension scheme generates samples with the variance that almost prohibit direct realization. Figure. 4.17 shows the available range of variance of the proposed system enhanced by the variance extension iterations.

4.3 Test Chip Measurement

The measured performance of our compact 2.05mm^2 LEIA chip at 0.9V in room temperature is compared with recently published ring-LWE implementations on different hardware platforms in Table. 4.2.

Table 4.2: Comparison with Previous Work

q	Distribution	Reference	n	Platform	Frequency	NTT Cycle	INTT Cycle	RNG Cycle
12289	DG[a],σ=215	*Oder, et al.* (2014)	512	ARM CortexM4	180MHz	508,624	508,624	935,925
		Liu, et al. (2017)	512	Atmel ATxmega128	32 MHz	516,971	468,090	105,153
			256			194,145	174,023	53,023
		This Work	**512**	**ASIC 40nm CMOS**	**300 MHz**	**492**	**572**	**7603.2**[b]
			256			**160**	**200**	**3801.6**
	Binomial, Φ_{16}	*Alkim, et al.* (2016)	512	ARM Cortex-M4	180MHz	87,223	97,789	54,332
		This Work		**ASIC 40nm CMOS**	**300MHz**	**492**	**572**	**3703.8**
7681	DG, $\sigma=\frac{11.32}{\sqrt{2\pi}}$	*Roy, et al.* (2014); *Verbauwhede, et al.* (2015)	256	Xilinx Virtex-6	313MHz	667	1048	805
				ASIC (synthesis) 130nm CMOS	500MHz			
		This Work		**ASIC 40nm CMOS**	**300MHz**	**160**	**200**	**262**

[a]Discrete Gaussian Distribution
[b]Average Performance over 100,000 Samples

LEIA is compatible with all the referenced schemes. LEIA's NTT accelerator demonstrates above 200x speedup over an NTT implementation on embedded processor and 4X speedup over a larger NTT implementation on FPGA. The core consumes 140mW at its operational mode which, given 200x speedup in cycle counts, demonstrates about 50X total energy saving compared to implementations on ARM platforms in *Alkim, et al.* (2016). LEIA's DG accelerator achieves more than 4x improvement over all previous results.

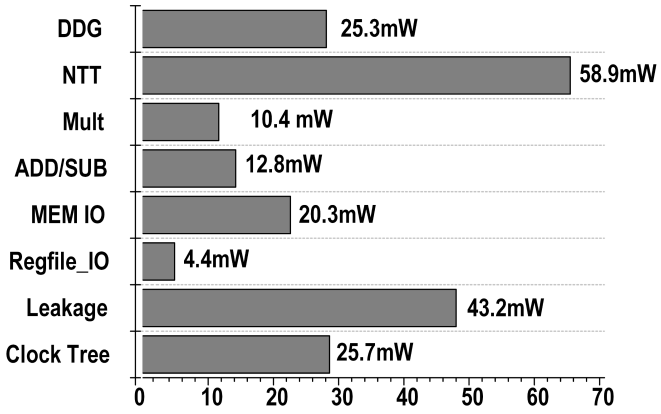
Another important aspect is that most of the previous ring-LWE implementations were designed for a narrow parameter range, while our design is compatible with all of them to support the widest range of applications.

Figure 4.18a and Table 4.3 show the power breakdown and key specs of the core. Figure 4.18b shows the chip photo.

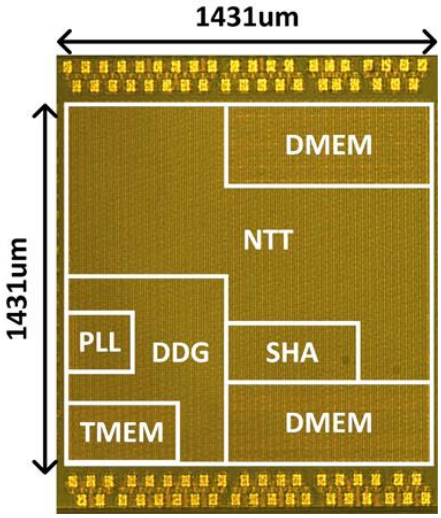
Table 4.3: Key Performance Spec

Technology	**40nm GP**
Core Frequency	**300 MHz**
Supply VDD	**0.9V**
Power[a]	**140mW (1 DDG Core)**
	216.5mW (4 DDG Core)
Area	**2.05 mm^2**
NTT Speed	**2.2MOp/S**
Vector Cache Size	**195kB**
Message Cache Size	**80kB**
DDG Table Max Size	**65kB**
Range of q	**2 to 2^{32}-1**
Range of n	**64 to 2048**
Range of σ	**1 to 1500**
RNG Scheme	**KY Sampling, ChaCha20**
Hash Table	**SHA-256,SHA-512**

[a]Benchmarked on NewHope*Alkim, et al.* (2016)



(a) Power Consumption for Each Operation



(b) LEIA Chip Photo

4.4 Conclusion

We present LEIA, the first silicon-proven RLWE accelerator. LEIA achieves more than 200X speedup and 50X better energy efficiency in key operations compared to ARM-based solutions. LEIA is also generic enough to be compatible with almost all

the RLWE based crypto-system realizations to date.

CHAPTER V

Conclusion

In this thesis, several new approaches have been proposed to address today's fast growing demand in wireline communication and security. In particular, two novel approaches have been proposed, analyzed and demonstrated to enable the energy-efficient bandwidth extension for wireline communications, and a new high-performance and efficient lattice encryption crypto accelerator is designed for post-quantum security applications.

As IP traffic volume on the internet grows exponentially towards the era of big data and cloud computing, demands on high speed serial interfaces between ICs have been increasing dramatically, putting stringent requirement on design solutions while creating opportunities to innovations. One significant requirement on the designs for wireline communications is the low power and area budget while maintaining a high data rate on the order of tens of Gb per second. Therefore, wireline transceivers operate sub-optimally on the system level, i.e. a distance away from Shannon's channel capacity. This work proposes design techniques that bring the system performance closer to the optimal while maintaining a low overall cost.

Meanwhile, the increased amount of data along with novel applications also pose new challenges in maintaining security and privacy of end users. Besides, the recent advancement of developing quantum computers has been bringing closer the need for

solutions for post-quantum security. Lattice-based cryptography is an efficient and versatile candidate solution to the challenges. However, practical implementations of lattice cryptosystems have mostly been based on processors and some on FPGAs. ASIC based solution provides much better performance, efficiency, and opportunities to exploit the many nice properties of lattice cryptography, offering support for a wide range of demanding applications.

On the optimal equalization aspect, a high throughput and low power maximum likelihood sequence detection (MLSD) based transceiver implementation is proposed and tested with a complete channel setup. MLSD offers the theoretically optimal BER given the sampled sequence of channel output. Previous MLSD implementations for high speed serial communications have been uncommon and are often restricted to optical channels due to the substantially higher complexity compared to filter based equalization systems. The proposed pipelined design is based on a M-Step formulation of the MLSD algorithm. The design exploits the speed of advanced technology nodes and achieves high data rate, and low cost on power and area, while providing flexible de-serialization factors.

In applications that demand low-cost equalization, a phase equalization approach is proposed. Typical equalizers operate on the magnitude of the signal spectrum, either amplifying the high frequency components like CTLE or directly canceling inter-symbol interference like DFE. SNR has almost always been traded off for bandwidth in these equalizers. Phase equalization, when applicable, enjoys the benefit of enhancing the main signal strength while suppressing ISI without losing SNR, due to the all-pass characteristics of the transfer function. In the proposed design, a simple RC-CR network based phase equalization scheme is proposed and analyzed. A complete receiver system including both clock and data paths have been designed as a proof of concept for the proposed technique. As a complete new way of performing equalization, the phase equalization has demonstrated competitive power and area

over the state-of-the-art.

Finally, an instruction based ASIC accelerator implementation, LEIA, for lattice-encryption has been designed and verified. The proposed architecture contains dedicated acceleration blocks for integer polynomial arithmetic and discrete noise vector generation. The polynomial arithmetic block utilizes number theoretic transform to achieve speedup with low power and area cost. The noise vector generator employs a Knuth-Yao sampler that achieves high precision, high speed and low overhead at the same time. The proposed design achieved 3 orders of magnitude higher speed and 4 orders of magnitude energy saving compared with recent embedded processor based solutions.

BIBLIOGRAPHY

BIBLIOGRAPHY

Ajtai, M. (1996), Generating hard instances of lattice problems, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 99–108, ACM.

Alkim, E., L. Ducas, T. Pöppelmann, and P. Schwabe (2016), Post-quantum key exchange-a new hope., in *USENIX Security Symposium*, pp. 327–343.

Alkim, et al., E. (2016), NewHope on ARM Cortex-M, in *SPACE*.

Anders, M. A., S. K. Mathew, S. K. Hsu, R. K. Krishnamurthy, and S. Borkar (2008), A 1.9 Gb/s 358 mW 16-256 State Reconfigurable Viterbi Accelerator in 90 nm CMOS, *IEEE Journal of Solid-State Circuits*, *43*(1), 214–222.

Bae, H.-M., J. Ashbrook, J. Park, N. Shanbhag, A. Singer, and S. Chopra (2006), An MLSE receiver for electronic-dispersion compensation of OC-192 fiber links, in *IEEE Int. Solid-State Circuits Conf.*, pp. 874–883, doi: 10.1109/ISSCC.2006.1696128.

Bai, R., S. Palermo, and P. Y. Chiang (2014), 2.5 a 0.25pj/b 0.7v 16gb/s 3-tap decision-feedback equalizer in 65nm cmos, in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 46–47, doi: 10.1109/ISSCC.2014.6757331.

Bergmans, J. (1996), *Digital Baseband Transmission and Recording*, Springer US.

Black, P., and T.-Y. Meng (1993), A 1 Gb/s, 4-state, sliding block Viterbi decoder, in *Proc. Symp. VLSI Circuits*, pp. 73–74, doi:10.1109/VLSIC.1993.920543.

Boesch, R., K. Zheng, and B. Murmann (2016), A 0.003 mm2 5.2 mw/tap 20 gbd inductor-less 5-tap analog rx-ffe, in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, pp. 1–2, doi:10.1109/VLSIC.2016.7573522.

Buchmann, J., D. Cabarcas, F. Göpfert, A. Hülsing, and P. Weiden (2013), Discrete ziggurat: A time-memory trade-off for sampling from a gaussian distribution over the integers, in *International Conference on Selected Areas in Cryptography*, pp. 402–417, Springer.

Cao, J., et al. (2010), A 500 mw ADC-based CMOS AFE with digital calibration for 10 Gb/s serial links over KR-backplane and multimode fiber, *IEEE. J. Solid-State Circuits*, *45*(6), 1172–1185.

Chen, E.-H., R. Yousry, and C.-K. Yang (2012), Power optimized ADC-based serial link receiver, *IEEE J. Solid-State Circuits*, *47*(4), 938–951, doi: 10.1109/JSSC.2012.2185356.

Cheng, C., R. Lu, A. Petzoldt, and T. Takagi (2017), Securing the internet of things in a quantum world, *IEEE Communications Magazine*, *55*(2), 116–120.

Cisco (2015), Cisco Global Cloud Index Forecast and Methodology 2015-2020, *White Paper*.

Cover, T. M., and J. A. Thomas (2012), *Elements of information theory*, John Wiley & Sons.

de Clercq, et al., R. (2015), Efficient software implementation of ring-LWE encryption, in *DATE*.

Ducas, L., V. Lyubashevsky, and T. Prest (2014), Efficient identity-based encryption over ntru lattices, in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 22–41, Springer.

Elahmadi, S., et al. (2010), An 11.1 Gbps analog PRML receiver for electronic dispersion compensation of fiber optic communications, *IEEE J. Solid-State Circuits*, *45*(7), 1330–1344, doi:10.1109/JSSC.2010.2049460.

Fettweis, G., and H. Meyr (1989), Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck, *IEEE Trans. Comm.*, *37*(8), 785–790, doi: 10.1109/26.31176.

Folláth, J. (2014), Gaussian sampling in lattice based cryptography, *Tatra Mountains Mathematical Publications*, *60*(1), 1–23.

Forney, J., G.D. (1973), The Viterbi algorithm, *IEEE Proc.*, *61*(3), 268–278, doi: 10.1109/PROC.1973.9030.

Gardner, F. (2005), *Phaselock Techniques*, Wiley.

Gentry, C., C. Peikert, and V. Vaikuntanathan (2008), Trapdoors for hard lattices and new cryptographic constructions, in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 197–206, ACM, New York, NY, USA, doi:10.1145/1374376.1374407.

Gondi, S., and B. Razavi (2007), Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers, *IEEE J. Solid-State Circuits*, *42*(9), 1999–2011, doi:10.1109/JSSC.2007.903076.

Hall, S., and H. Heck (2009), *Advanced Signal Integrity for High-Speed Digital Designs*, Wiley.

Hanumolu, P. K., G.-Y. Wei, and U.-K. Moon (2005), Equalizers for high-speed serial links, *Int. J. High Speed Electron. and Syst.*, *15*(02), 429–458.

Hoffstein, J., J. Pipher, and J. H. Silverman (1998), Ntru: A ring-based public key cryptosystem, in *International Algorithmic Number Theory Symposium*, pp. 267–288, Springer.

Horowitz, M., C.-K. K. Yang, and S. Sidiropoulos (1998), High-speed electrical signaling: overview and limitations, *IEEE Micro*, *18*(1), 12–24, doi:10.1109/40.653013.

Howe, J., C. Moore, M. O'Neill, F. Regazzoni, T. Gneysu, and K. Beeden (2016), Lattice-based encryption over standard lattices in hardware, in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–6, doi:10.1145/2897937.2898037.

Hu, S., H. Krll, Q. Huang, and F. Rusek (2017), Optimal channel shortener design for reduced-state soft-output Viterbi equalizer in single-carrier systems, *IEEE Trans. Commun.*, *65*(6), 2568–2582, doi:10.1109/TCOMM.2017.2685380.

Kermani, M. M., V. Singh, and R. Azarderakhsh (2017), Reliable low-latency Viterbi algorithm architectures benchmarked on ASIC and FPGA, *IEEE Trans. Circuits Syst. I, Reg. Papers*, *64*(1), 208–216, doi:10.1109/TCSI.2016.2610187.

Khedr, A., G. Gulak, and V. Vaikuntanathan (2016), Shield: scalable homomorphic implementation of encrypted data-classifiers, *IEEE Transactions on Computers*, *65*(9), 2848–2858.

L'Heureux, A., K. Grolinger, H. F. ElYamany, and M. Capretz (2017), Machine learning with big data: Challenges and approaches, *IEEE Access*.

Lin, I.-C., and T.-C. Liao (2017), A survey of blockchain security issues and challenges., *IJ Network Security*, *19*(5), 653–659.

Liu, J., and X. Lin (2004a), Equalization in high-speed communication systems, *IEEE Circuits and Systems Magazine*, *4*(2), 4–17, doi:10.1109/MCAS.2004.1330746.

Liu, J., and X. Lin (2004b), Equalization in high-speed communication systems, *IEEE Circuits Syst. Mag.*, *4*(2), 4–17, doi:10.1109/MCAS.2004.1330746.

Liu, et al. (2017), High-performance ideal lattice-based cryptography on 8-bit AVR microcontrollers, *ACM Trans. Embed. Comput. Syst.*

Lyubashevsky, V., C. Peikert, and O. Regev (2010), *On Ideal Lattices and Learning with Errors over Rings*, pp. 1–23, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-13190-51.

Lyubashevsky, V., C. Peikert, and O. Regev (2013), On ideal lattices and learning with errors over rings, *J. ACM*, *60*(6), 43:1–43:35, doi:10.1145/2535925.

Matsumoto, T., and H. Imai (1988), *Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption*, pp. 419–453, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/3-540-45961-839.

Merkle, R. C. (1989), A certified digital signature, in *Conference on the Theory and Application of Cryptology*, pp. 218–238, Springer.

Micciancio, D., and O. Regev (2009), Lattice-based cryptography, in *Post-quantum cryptography*, pp. 147–191, Springer.

Micciancio, D., and M. Walter (2017), Gaussian sampling over the integers: Efficient, generic, constant-time., *IACR Cryptology ePrint Archive*, *2017*, 259.

Micciancio, et al. (2017), Gaussian sampling over the integers: Efficient, generic, constant-time, in *CRYPTO*, Springer International Publishing.

Mohseni, M., et al. (2017), Commercialize early quantum technologies.

Monz, T., D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt (2016), Realization of a scalable shor algorithm, *Science*, *351*(6277), 1068–1070.

Oder, et al., T. (2014), Beyond ECDSA and RSA: Lattice-based digital signatures on constrained devices, in *DAC*, pp. 1–6.

Overbeck, R., and N. Sendrier (2009), Code-based cryptography, in *Post-quantum cryptography*, pp. 95–145, Springer.

Palermo, S. (2011), *CMOS Nanoelectronics Analog and RF VLSI Circuits. Chapter 9: High-Speed Serial I/O Design for Channel-Limited and Power-Constrained Systems*, McGraw-Hill.

Palermo, S. (2017), *ECEN 720 Lectures*.

Patterson, D. A., and J. L. Hennessy (2017), *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*, Morgan kaufmann.

Pavan, S., R. Schreier, and G. Temes (2016), *Understanding Delta-Sigma Data Converters*, IEEE Press Series on Microelectronic Systems, Wiley.

Peng, H., R. Liu, Y. Hou, and L. Zhao (2016), A Gb/s parallel block-based Viterbi decoder for convolutional codes on GPU, in *Int. Conf. Wireless Commun. Signal Process.*, pp. 1–6, doi:10.1109/WCSP.2016.7752638.

Pernillo, J., and M. Flynn (2011), A 1.5-GS/s flash ADC with 57.7-dB SFDR and 6.4-bit ENOB in 90 nm digital CMOS, *IEEE Trans. Circuits Syst. II, Exp. Briefs*, *58*(12), 837–841, doi:10.1109/TCSII.2011.2168020.

Razavi, B. (2012), *RF Microelectronics*, Prentice Hall Communications E, Prentice Hall.

Regev, O. (2009), On lattices, learning with errors, random linear codes, and cryptography, *Journal of the ACM (JACM)*, *56*(6), 34.

Regev, O. (2010), The learning with errors problem, *Invited survey in CCC*, p. 15.

Roy, S. S., F. Vercauteren, and I. Verbauwhede (2013), High precision discrete gaussian sampling on fpgas, in *International Conference on Selected Areas in Cryptography*, pp. 383–401, Springer.

Roy, et al. (2014), Compact ring-LWE cryptoprocessor, in *CHES*, pp. 371–391, Springer-Verlag New York, Inc.

Salehi, M., and J. Proakis (2007), *Digital Communications*, McGraw-Hill Education.

Shafik, A., E. Z. Tabasy, S. Cai, K. Lee, S. Hoyos, and S. Palermo (2016), A 10 gb/s hybrid adc-based receiver with embedded analog and per-symbol dynamically enabled digital equalization, *IEEE Journal of Solid-State Circuits*, *51*(3), 671–685.

Shannon, C. E. (2001), A mathematical theory of communication, *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55.

Ting, C., J. Liang, A. Sheikholeslami, M. Kibune, and H. Tamura (2013), A blind baud-rate ADC-based CDR, in *IEEE Int. Solid-State Circuits Conf.*, pp. 122–123, doi:10.1109/ISSCC.2013.6487664.

Toifl, T., C. Menolfi, P. Buchmann, M. Kossel, T. Morf, R. Reutemann, M. Ruegg, M. L. Schmatz, and J. Weiss (2005), A 0.94-ps-rms-jitter 0.016-mm2 2.5-ghz multiphase generator pll with 360 deg; digitally programmable phase shift for 10-gb/s serial links, *IEEE Journal of Solid-State Circuits*, *40*(12), 2700–2712, doi: 10.1109/JSSC.2005.856581.

Veigel, T., T. Alpert, F. Lang, M. Grzing, and M. Berroth (2013), A Viterbi Equalizer Chip for 40 Gb/s optical communication links, in *2013 European Microwave Integrated Circuit Conference*, pp. 49–52.

Verbauwhede, et al., I. (2015), 24.1 circuit challenges from cryptography, in *ISSCC*.

Wang, Y., and B. V. K. V. Kumar (2017), Improved multitrack detection with hybrid 2D equalizer and modified Viterbi detector, *IEEE Trans. Magn.*, *PP*(99), 1–1, doi: 10.1109/TMAG.2017.2708687.

Yang, Y., L. Wu, G. Yin, L. Li, and H. Zhao (2017), A survey on security and privacy issues in internet-of-things, *IEEE Internet of Things Journal*, *4*(5), 1250–1258, doi: 10.1109/JIOT.2017.2694844.

Yueksel, H., G. Cherubini, R. D. Cideciyan, A. Burg, and T. Toifl (2016a), Design considerations on sliding-block Viterbi detectors for high-speed data transmission, in *Int. Conf. Signal Process. and Commun.*, pp. 1–6, doi: 10.1109/ICSPCS.2016.7843366.

Yueksel, H., et al. (2016b), A 4.1 pJ/b 25.6 Gb/s 4-PAM reduced-state sliding-block Viterbi detector in 14 nm CMOS, in *European Solid-State Circuits Conf.*, pp. 309–312, doi:10.1109/ESSCIRC.2016.7598304.

Zhang, B., A. Nazemi, A. Garg, N. Kocaman, M. Ahmadi, M. Khanpour, H. Zhang, J. Cao, and A. Momtaz (2013), A 195mw / 55mw dual-path receiver AFE for multistandard 8.5-to-11.5 Gb/s serial links in 40nm CMOS, in *IEEE Int. Solid-State Circuits Conf.*, pp. 34–35, doi:10.1109/ISSCC.2013.6487625.

Zhou, Q., W. Wang, Y. Yao, and Z. Chen (2017), A qaud_byte implementation of 8b10b decoder in jesd204b protocol, in *Intelligent Robot Systems (ACIRS), 2017 2nd Asia-Pacific Conference on*, pp. 137–140, IEEE.