

# A Maximum-Likelihood Sequence Detection Powered ADC-Based Serial Link

Shiming Song<sup>1</sup>, *Student Member, IEEE*, Kyojin D. Choo, *Member, IEEE*, Thomas Chen, *Student Member, IEEE*, Sunmin Jang, *Student Member, IEEE*, Michael P. Flynn, *Fellow, IEEE*, and Zhengya Zhang<sup>1</sup>, *Member, IEEE*

**Abstract**—A 0.88 mm<sup>2</sup> 65-nm analog-to-digital converter (ADC)-based serial link transceiver is designed with a maximum-likelihood sequence detector (MLSD) for robust equalization. The MLSD is optimized in a pipelined look-ahead architecture to reach 10 Gb/s at 5.8 pJ/b and 5 Gb/s at 3.9 pJ/b, making it practical for an energy-efficient ADC-based serial link. Compared with linear equalizer and decision feedback equalizer, the MLSD provides extra margin to accommodate timing offsets, ADC nonlinearities, and voltage noise, which is exploited by co-designing the analog front-end to reduce its power and area. We present a 2x-oversampled and 2-way interleaved 5 b stochastic flash ADC architecture. No front-end analog equalizer, buffer or sample, and hold amplifier are needed. Tested with a 45-cm FR-4 trace, the serial link transceiver achieves 5 Gb/s at a bit error rate below 10<sup>-11</sup> with a 7% UI margin without any analog front-end equalization, consuming 54.5 mW in receiver and 16.2 mW in transmitter.

**Index Terms**—ML detection, Viterbi detector, equalizer, serial link.

## I. 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) [1]–[5]. 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 [6]. 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 [7]–[12]. In [7] and [8], a simplified version of MLSD was implemented and verified on an emulated channel targeting 100Gb/s Ethernet. Kermani *et al.* [9] 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., [10]–[12]. However, conventional multi-Gb/s MLSDs consume on the order of 100pJ/b [13]–[17], 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 analog-to-digital converter (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 interpolator. Our key contributions include an efficient, high-speed MLSD architecture inspired by [18], 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 II. The serial nature

Manuscript received June 21, 2017; revised September 19, 2017 and November 3, 2017; accepted November 5, 2017. This work was supported in part by NSF under Grant CCF-1255702, in part by SRC, and in part by the NSF Graduate Research Fellowship Program. This paper was recommended by Associate Editor A. Nagari. (*Corresponding author: Shiming Song.*)

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: shisong@umich.edu; kjchoo@umich.edu; tcchen@umich.edu; smjang@umich.edu; mpflynn@umich.edu; zhengya@umich.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSI.2017.2775619

of the Viterbi algorithm makes it challenging to design a high-throughput MLSD. We present a reformulation of the Viterbi algorithm in Section III, 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 IV, 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 V before the conclusion of this paper.

## II. MLSD EQUALIZATION THEORY

Channel distortion places bandwidth limitations that show up as ISI in time domain. With linearity assumptions, which generally holds for channels consisting of passive components, one commonly models the channel as (1) [19].

$$\mathbf{y}_i = \sum_{j=-\infty}^{\infty} \mathbf{x}_j * \mathbf{h}_{i-j} + \mathbf{n}_i, \quad (1)$$

where  $\mathbf{y}$  is the observed channel output at the  $i$ -th time step from the ADC,  $\mathbf{x}$  is the modulated transmitter output, i.e.,  $+1$ 's and  $-1$ 's in the binary case,  $\mathbf{h}$  is the sampled channel response to a single pulse [20] and  $\mathbf{n}$  is the sampled noise process, usually assumed to be Gaussian. A widely accepted optimal detection technique is MLSD [6] that directly minimizes the probability of error. The MLSD under Gaussian noise and binary transmission assumption takes the form shown in (2).

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

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 length become finite or approximately finite. Direct brute force search for a solution to (2) would take prohibitive computation cost, on the order of  $2^N$ , where  $N$  is the block length.

### A. 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. 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  $\nu$ , is defined as the length of channel memory, e.g., in this case  $\nu = 3$ . For a channel of constraint length  $\nu$ , the channel response at a given time point depends on the current bit and also the  $\nu - 1$  bits that are transmitted immediately prior to this time point. For example, the two post-cursor taps shown in Fig. 1(a) indicate that the channel response at a given time point depends on not only

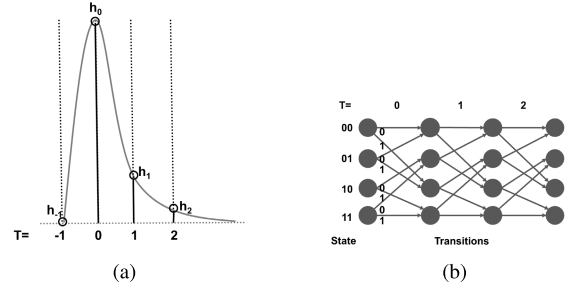


Fig. 1. Pulse response and its representation in a trellis diagram. (a) Example of sampled pulse response of the channel. (b) Corresponding trellis diagram.

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  $2^{\nu-1}$  possible combinations of the post-cursor bits. In a trellis representation, each combination forms a state. Fig. 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. 1.

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

$$\mathbf{S}_i = (\mathbf{x}_i, \mathbf{x}_{i-1}, \dots, \mathbf{x}_{i-\nu+2}). \quad (3)$$

Note that we assume both sequences are infinitely long at this point for simplicity in indexing. The state at each time step consists of  $\nu - 1$  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 (3) and the inverse defined as

$$\mathbf{x}_i = \mathbf{S}_i[1], \quad \mathbf{x}_{i-1} = \mathbf{S}_{i-1}[1], \dots, \quad (4)$$

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. 2.

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

$$\gamma(\mathbf{S}_i, \mathbf{S}_{i+1}) = (\mathbf{y}_{i+1} - \sum_{j=-\infty}^{\infty} \mathbf{x}_j * \mathbf{h}_{i+1-j})^2, \quad (5)$$

which is essentially a term in the summation in (2). Therefore the maximum likelihood (ML) optimization (2) can be rewritten as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{S}} \sum_{i=-\infty}^{\infty} \gamma(\mathbf{S}_i, \mathbf{S}_{i+1}). \quad (6)$$

Observing that each possible state sequence  $\mathbf{S}$  is also a path on the trellis diagram, and (6) reduces the original ML problem (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.

### B. 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  $\mathbf{S}_i$ .

$$PM_{i,T} = \min_{\mathbf{S}_{i-1}} \sum_{k=-\infty}^i \gamma(\mathbf{S}_{k-1}, \mathbf{S}_k). \quad (7)$$

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 1, can be applied to significantly simplify the path metrics computation.

*Theorem 1 (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

$$PM_{i-1, \mathbf{S}_{i-1}} + \gamma(\mathbf{S}_{i-1}, T) < PM_{i-1, \mathbf{S}'_{i-1}} + \gamma(\mathbf{S}'_{i-1}, T) \quad (8)$$

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

Theorem (Thm). 1 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 [15]. 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 [15].

## III. 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 [14], [15] 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.

### A. Matrix Formulation of Viterbi Algorithm

To arrive at a more efficient high-throughput architecture, we look at the matrix formulation of the Viterbi algorithm [18]. 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. 3.

*Definition 3 (Add and Multiply Operations):* On the real numbers,  $a, b \in \mathbb{R}$ , a pair of add and multiply operations can be defined as

$$a \boxplus b = \min(a, b) \text{ (Add)} \quad (9)$$

$$a \boxtimes b = a + b \text{ (Multiply)}. \quad (10)$$

It can be shown that the set of real numbers together with these two operations form a semi-ring [18], 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

$$\mathbf{C}_i = \mathbf{M}_i \mathbf{C}_{i-1}. \quad (11)$$

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 (11) as

$$\mathbf{C}_N = \left( \prod_{i=-\infty}^N \mathbf{M}_i \right) \mathbf{C}_0, \quad (12)$$

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 2 (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

$$\sum_{k=i}^{j-1} \gamma(\mathbf{S}_k, \mathbf{S}_{k+1}) < \sum_{k=i}^{j-1} \gamma(\mathbf{S}'_k, \mathbf{S}'_{k+1}) \quad (13)$$

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

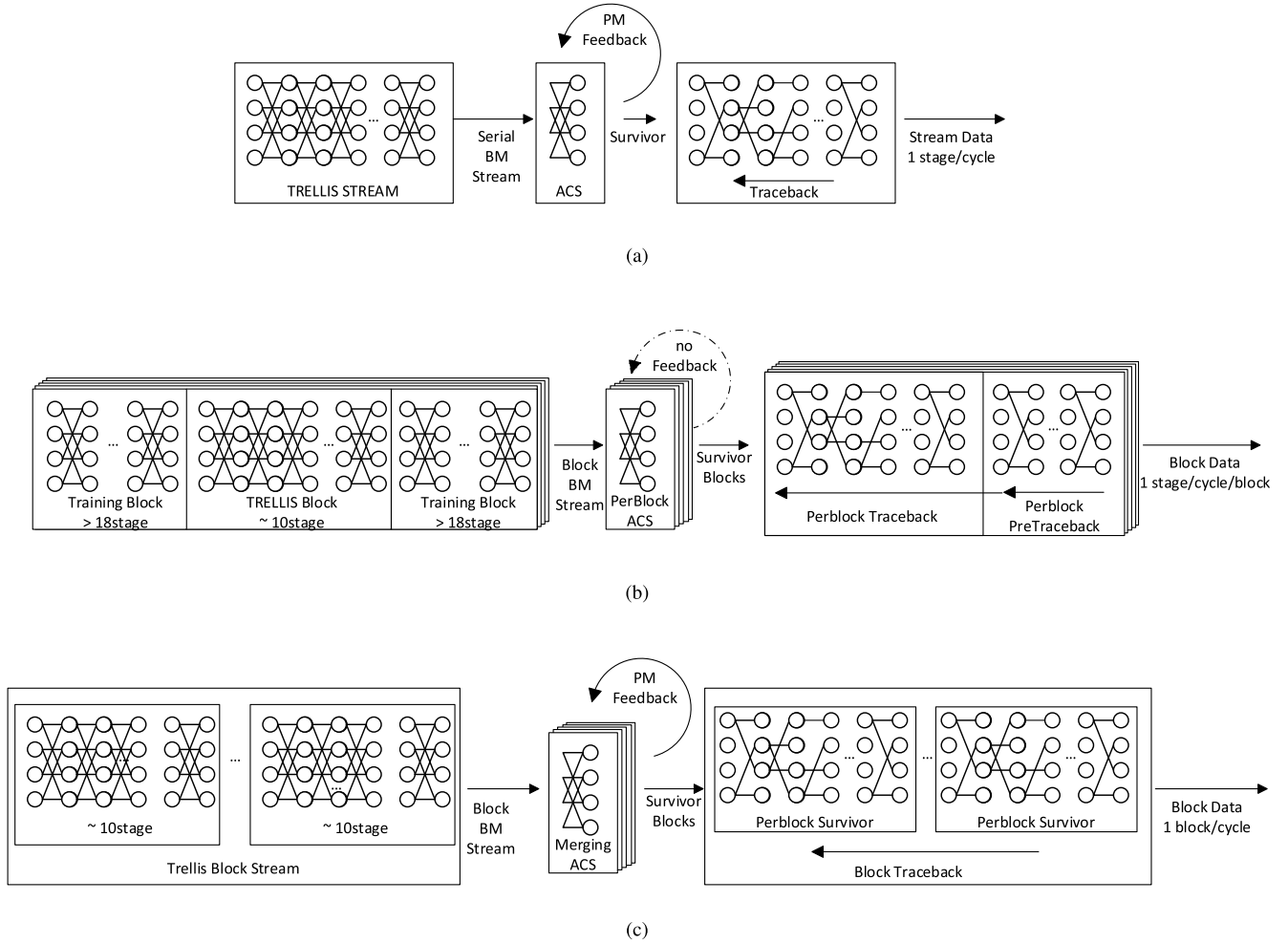


Fig. 2. MLSD architectures: (a) serial architecture; (b) sliding block architecture; and (c) pipelined look-ahead architecture.

The matrix-matrix multiplication used in (12) is just a result of the direct application of Thm. 2. 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 (12). In the following section, we discuss our implementation based on the *Generalized Principle of Optimality*.

### B. Efficient and High-Throughput MLSD Architecture

A serial MLSD architecture based on (11) is illustrated in Fig. 2a. 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  $\mathcal{O}(N)$ . 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 (11) is shown in Fig. 2b. 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 [15]. However, the sliding block architecture requires a long pre-training frame on each side, and it can become an excessive overhead. In the example shown in Fig. 2b with a constraint length of  $v = 3$ , 36 training stages are required in total to decode a block.

Inspired by the aforementioned matrix formulation of the Viterbi algorithm [18], we propose an alternative serial MLSD architecture based on (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. 2c. 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 pipelined 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 that is  $1/P$  of the sample rate. For flexibility in choosing  $P$  and not being bound by the power of 2 requirement, we use a pipelined look-ahead architecture in implementing the MLSD.

### C. 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 simulation shows that a 4-tap MLSD running at 5Gb/s provides a 24ps timing margin at  $10^{-8}$  bit error rate (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. 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 correction of the AFE.

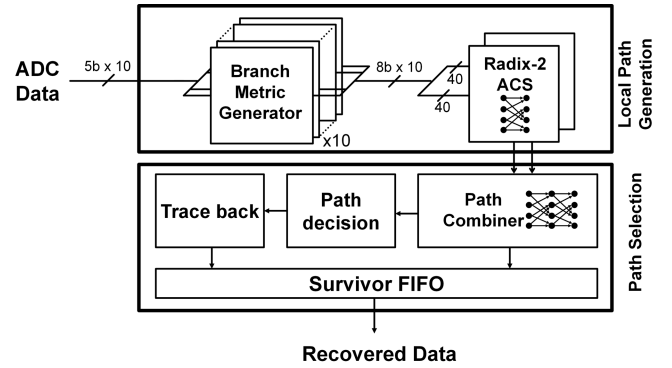


Fig. 3. Pipelined look-ahead MLSD implementation.

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. 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.

To summarize, we use several methods to improve power efficiency of the MLSD, as illustrated in Fig. 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 [15] is also shown in Fig. 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.

### IV. 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 analyses based on [20] show that

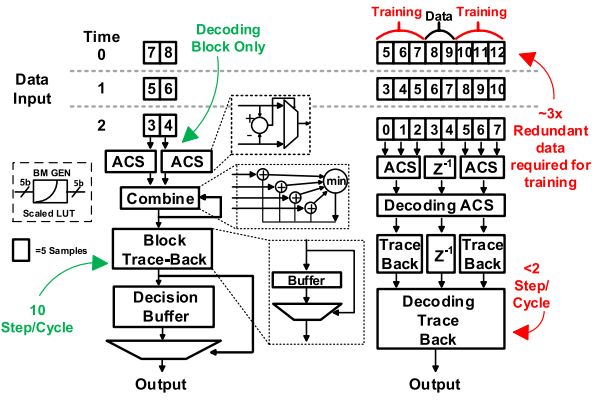


Fig. 4. Comparison of the pipelined look-ahead architecture (left) and the sliding block architecture (right).

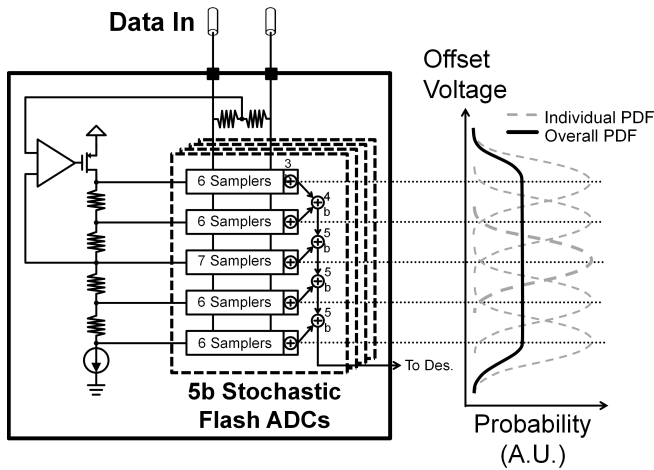


Fig. 5. Stochastic flash ADC design.

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.

#### A. 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 [21] to keep the power consumption and input capacitance low compared to conventional flash ADCs [22], [23]. The 2x-oversampled and 4-way interleaved ADC shown in Fig. 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. In our ADC design, we use StrongArm comparators, which are fast and give reasonably large random offsets. For the ADC to cover the full signal range, the comparators are grouped and

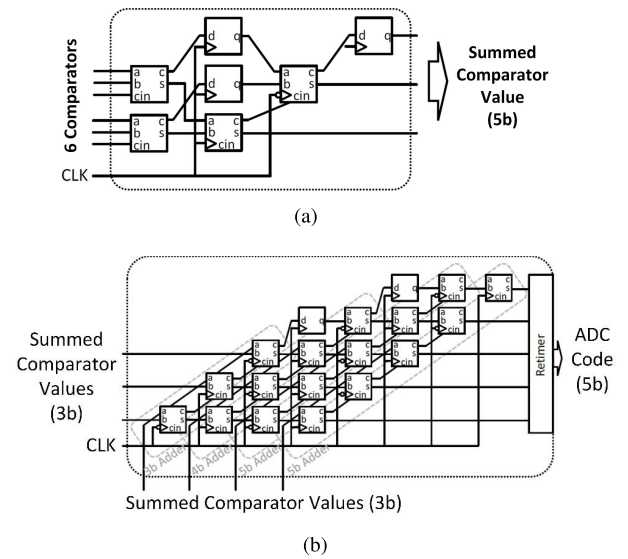


Fig. 6. Adder structures used in stochastic ADC. (a) Adder structure for local summation. (b) Adder structure for final summation.

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 devices are extensively used in the first stage of the comparator, to make it run at higher speed.

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. 6a. These values are then summed using structure shown in Fig. 6b 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.

#### B. 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. 7, that generates 32 phases from the 8 phases produced by the on-chip PLL.

The digital clock recovery loop selects the phase that best represents the center of the unit interval (UI), as shown in Fig. 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

The overall architecture of our prototype 5Gb/s ADC-based serial-link transceiver with MLSD is shown in Fig. 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.

TABLE I  
COMPARISON WITH PREVIOUS ADC-BASED LINKS

	This Work	[24]	[22]	[23]	[25]	[26]
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 <sup>2</sup> )	0.14	0.185	3	0.25	0.82	0.38
EQ Structure	3-tap MLSD	5-Tap DFE	FFE+DFE	2-Tap DFE	FFE+DFE	FFE+DFE
EQ Power (mW)	19.3	37	-	111	-	10
EQ Area (mm <sup>2</sup> )	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 <sup>2</sup> )	0.35	0.26	15	0.536	-	0.81

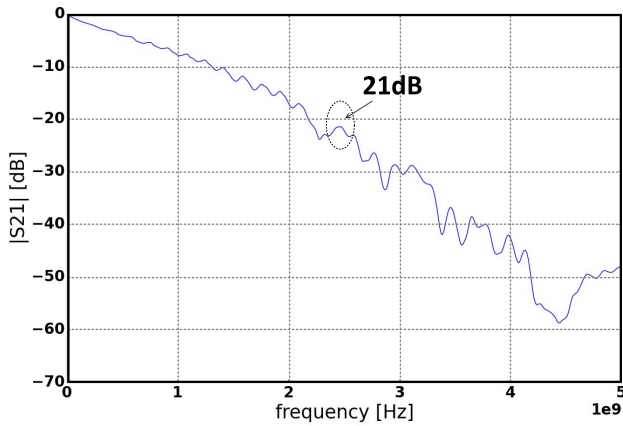


Fig. 12. Insertion loss of the test FR-4 trace.

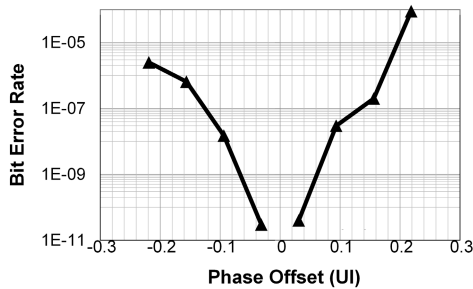


Fig. 13. Test setup bathtub curve.

of magnitude. The MLSDs are compared based on 3-tap configuration.

Transmit and receive operation are verified at 5Gb/s. The transmitter 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,

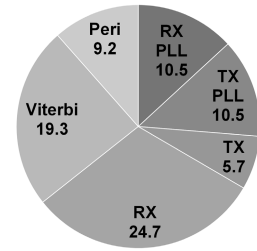


Fig. 14. Test chip power measurements (mW).

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, as shown in Fig. 12, and testing shows that BER under this condition is better than  $10^{-11}$ . The measured bathtub curve is shown in Fig. 13.

The power breakdown is presented in Fig. 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 [22]–[26] in Table. I. 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.

## VI. 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 exceeds the state-of-the-art multi-Gb/s MLSDs by over an order of magnitude.

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<sup>2</sup>. The design achieves a competitive FoM of 10.9mW/Gb/s.

## REFERENCES

- [1] S. Palermo, *CMOS Nanoelectronics: Analog and RF VLSI Circuits: High-Speed Serial I/O Design for Channel-Limited and Power-Constrained Systems*. New York, NY, USA: McGraw-Hill, 2011, ch. 9.
- [2] M. Horowitz, C.-K. K. Yang, and S. Sidiropoulos, "High-speed electrical signaling: Overview and limitations," *IEEE Micro*, vol. 18, no. 1, pp. 12–24, Jan./Feb. 1998.
- [3] P. K. Hanumolu, G.-Y. Wei, and U.-K. Moon, "Equalizers for high-speed serial links," *Int. J. High Speed Electron. Syst.*, vol. 15, no. 2, pp. 429–458, Jun. 2005.
- [4] J. Liu and X. Lin, "Equalization in high-speed communication systems," *IEEE Circuits Syst. Mag.*, vol. 4, no. 2, pp. 4–17, 2004.
- [5] S. Gondi and B. Razavi, "Equalization and clock and data recovery techniques for 10-Gb/s CMOS serial-link receivers," *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 1999–2011, Sep. 2007.
- [6] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [7] H. Yueksel *et al.*, "A 4.1 pJ/b 25.6 Gb/s 4-PAM reduced-state sliding-block Viterbi detector in 14 nm CMOS," in *Proc. Eur. Solid-State Circuits Conf.*, Sep. 2016, pp. 309–312.
- [8] H. Yueksel, G. Cherubini, R. D. Cideciyan, A. Burg, and T. Toifl, "Design considerations on sliding-block Viterbi detectors for high-speed data transmission," in *Proc. Int. Conf. Signal Process. Commun.*, Dec. 2016, pp. 1–6.
- [9] M. M. Kermani, V. Singh, and R. Azarderakhsh, "Reliable low-latency Viterbi algorithm architectures benchmarked on ASIC and FPGA," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 1, pp. 208–216, Jan. 2017.
- [10] S. Hu, H. Kröll, Q. Huang, and F. Rusek, "Optimal channel shortener design for reduced-state soft-output Viterbi equalizer in single-carrier systems," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2568–2582, Jun. 2017.
- [11] Y. Wang and B. V. K. Vijaya Kumar, "Improved multitrack detection with hybrid 2-D equalizer and modified Viterbi detector," *IEEE Trans. Magn.*, vol. 53, no. 10, May 2017, Art. no. 3000710.
- [12] H. Peng, R. Liu, Y. Hou, and L. Zhao, "A Gb/s parallel block-based Viterbi decoder for convolutional codes on GPU," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Oct. 2016, pp. 1–6.
- [13] S. Elahmadi *et al.*, "An 11.1 Gbps analog PRML receiver for electronic dispersion compensation of fiber optic communications," *IEEE J. Solid-State Circuits*, vol. 45, no. 7, pp. 1330–1344, Jul. 2010.
- [14] H.-M. Bae, J. Ashbrook, J. Park, N. Shanbhag, A. Singer, and S. Chopra, "An MLSE receiver for electronic-dispersion compensation of OC-192 fiber links," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2006, pp. 874–883.
- [15] P. Black and T.-Y. Meng, "A 1 Gb/s, 4-state, sliding block Viterbi decoder," in *Proc. Symp. VLSI Circuits*, May 1993, pp. 73–74.
- [16] M. A. Anders, S. K. Mathew, S. K. Hsu, R. K. Krishnamurthy, and S. Borkar, "A 1.9 Gb/s 358 mW 16–256 state reconfigurable Viterbi accelerator in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 43, no. 1, pp. 214–222, Jan. 2008.
- [17] T. Veigel, T. Alpert, F. Lang, M. Grözing, and M. Berroth, "A Viterbi equalizer chip for 40 Gb/s optical communication links," in *Proc. Eur. Microw. Integr. Circuit Conf.*, Oct. 2013, pp. 49–52.
- [18] G. Fettweis and H. Meyr, "Parallel Viterbi algorithm implementation: Breaking the ACS-bottleneck," *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 785–790, Aug. 1989.
- [19] M. Salehi and J. Proakis, *Digital Communications*. New York, NY, USA: McGraw-Hill, 2007. [Online]. Available: <https://books.google.com/books?id=HroiQAAACAAJ>
- [20] S. H. Hall and H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. Hoboken, NJ, USA: Wiley, 2009. [Online]. Available: <https://books.google.com/books?id=AB2DHvhSHpsC>
- [21] J. Pernillo and M. Flynn, "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*, vol. 58, no. 12, pp. 837–841, Dec. 2011.
- [22] J. Cao *et al.*, "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*, vol. 45, no. 6, pp. 1172–1185, Jun. 2010.
- [23] C. Ting, J. Liang, A. Sheikholeslami, M. Kibune, and H. Tamura, "A blind baud-rate ADC-based CDR," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013, pp. 122–123.
- [24] E.-H. Chen, R. Yousry, and C.-K. K. Yang, "Power optimized ADC-based serial link receiver," *IEEE J. Solid-State Circuits*, vol. 47, no. 4, pp. 938–951, Apr. 2012.
- [25] B. Zhang *et al.*, "A 195 mW/55 mW dual-path receiver AFE for multistandard 8.5-to-11.5 Gb/s serial links in 40 nm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2013, pp. 34–35.
- [26] A. Shafik, E. Z. Tabasy, S. Cai, K. Lee, S. Hoyos, and S. Palermo, "A 10 Gb/s hybrid ADC-based receiver with embedded analog and per-symbol dynamically enabled digital equalization," *IEEE J. Solid-State Circuits*, vol. 51, no. 3, pp. 671–685, Mar. 2016.

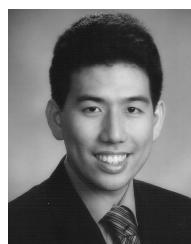


**Shiming Song** (S'13) received the B.S. degree from the University of Michigan, Ann Arbor, where he is currently pursuing Ph.D. degree in electrical engineering. In 2015, he did an internship at Texas Instruments, Santa Clara, where he was involved in fractional-N PLL system and circuit designs. His research interests include wireline transceiver design, channel coding, information theory, DSP, cryptography, and scientific computing.



**Kyojin D. Choo** (S'13–M'14) received the B.S. and M.S. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2007 and 2009, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Michigan.

From 2009 to 2013, he was with the Image Sensor Development Team, Samsung Electronics, Kiheung, South Korea, where he was involved in developing analog/mixed-signal readout circuits. He was involved in high-speed serial links, PLL/DLLs, and precision analog circuits. His current research interests include low-power sensor design, mm-scale system integration, and low-noise timing generation circuits.



**Thomas Chen** (S'15) received the B.S. and M.S. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

He did an internship with the Circuits Research Lab, Intel Corporation, in 2015. His research interests are high-speed and low-power VLSI circuits and systems. He received the Rackham Merit Fellowship from the University of Michigan in 2013 and the NSF Graduate Research Fellowship in 2015.



**Sunmin Jang** was born in Seoul, South Korea, in 1987. He received the B.S. degree in electrical engineering from Seoul National University, Seoul, in 2013, and the M.S. degree in electrical engineering from the University of Michigan in 2015, where he is currently pursuing the Ph.D. degree with a focus on digital beamformers. He was a recipient of the Samsung Scholarship in 2015.



**Michael P. Flynn** (M'95–SM'98–F'15) received the Ph.D. degree from Carnegie Mellon University in 1995. From 1988 to 1991, he was with the National Microelectronics Research Centre, Cork, Ireland. He was with National Semiconductor, Santa Clara, CA, USA, from 1993 to 1995. From 1995 to 1997, he was a member of Technical Staff with Texas Instruments, Dallas, TX, USA. During the four-year period from 1997 to 2001, he was with Parthus Technologies, Cork. He joined the University of Michigan in 2001, where he is currently a

Professor. His technical interests are in RF circuits, data conversion, serial transceivers and biomedical systems.

Dr. Flynn is a 2008 Guggenheim Fellow. He received the 2016 University of Michigan Faculty Achievement Award. He received the 2011 Education Excellence Award and the 2010 College of Engineering Ted Kennedy Family Team Excellence Award from the College from Engineering at the University of Michigan. He received the 2005–2006 Outstanding Achievement Award from the Department of Electrical Engineering and Computer Science at the University of Michigan. In 2004, he received the NSF Early Career Award. He was a recipient of the 1992–1993 IEEE Solid-State Circuits Pre-Doctoral Fellowship.

Dr. Flynn was the Editor-in-Chief of the IEEE JOURNAL OF SOLID STATE CIRCUITS (JSSC) from 2013 to 2016. He is a former Distinguished Lecturer of the IEEE Solid-State Circuits Society. He served as Associate Editor of the IEEE JSSC and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He serves on the Technical Program Committees of the International Solid State Circuits Conference and the European Solid State Circuits Conference. He served on the Technical Program Committees the Asian Solid-State Circuits Conference and the Symposium on VLSI Circuits.



**Zhengya Zhang** (S'02–M'09) received the B.A.Sc. degree in computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2003, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 2005 and 2009, respectively.

He has been a Faculty Member with the University of Michigan, Ann Arbor, MI, USA, since 2009, where he is currently an Associate Professor with the Department of Electrical Engineering and

Computer Science. His current research interests include low-power and high-performance VLSI circuits and systems for computing, communications, and signal processing.

Dr. Zhang was a recipient of the National Science Foundation CAREER Award in 2011, the Intel Early Career Faculty Award in 2013, the David J. Sakrison Memorial Prize for Outstanding Doctoral Research in electrical engineering and computer sciences at UC Berkeley, and the Best Student Paper Award at the Symposium on VLSI Circuits. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS (2013–2015) and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS (2014–2015). He has been an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS since 2015.