

# Research Report

## Design Techniques for High-Speed Multi-Level Viterbi Detectors and Trellis-Coded-Modulation Decoders

Hazar Yueksel<sup>1,2,4</sup>, Matthias Braendli<sup>1</sup>, Andreas Burg<sup>2</sup>, Giovanni Cherubini<sup>1</sup>, Roy D. Cideciyan<sup>1</sup>, Pier Andrea Francese<sup>1</sup>, Simeon Furrer<sup>1</sup>, Marcel Kossel<sup>1</sup>, Lukas Kull<sup>1</sup>, Danny Luu<sup>1,3</sup>, Christian Menolfi<sup>1</sup>, Thomas Morf<sup>1</sup>, Thomas Toifl<sup>1</sup>

<sup>1</sup>IBM Research – Zurich  
8803 Rüschlikon  
Switzerland

<sup>2</sup>École Polytechnique Fédérale de Lausanne  
1015 Lausanne  
Switzerland

<sup>3</sup>ETH Zurich  
8092 Zürich  
Switzerland

<sup>4</sup>Now at IBM T. J. Watson Research Center, NY 10598, USA

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This is the accepted version of the article published by IEEE: H. Yueksel, M. Braendli, A. Burg, G. Cherubini, R. D. Cideciyan, P. A. Francese, S. Furrer, M. Kossel, L. Kull, D. Luu, C. Menolfi, T. Morf, T. Toifl, "Design Techniques for High-Speed Multi-Level Viterbi Detectors and Trellis-Coded-Modulation Decoders," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 10, pp. 3529-3542, Oct. 2018.

[doi: 10.1109/TCSI.2018.2803735](https://doi.org/10.1109/TCSI.2018.2803735)

### LIMITED DISTRIBUTION NOTICE

This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies (e.g., payment of royalties). Some reports are available at <http://domino.watson.ibm.com/library/Cyberdig.nsf/home>.



Research

Africa • Almaden • Austin • Australia • Brazil • China • Haifa • India • Ireland • Tokyo • Watson • Zurich

# Design Techniques for High-Speed Multi-Level Viterbi Detectors and Trellis-Coded-Modulation Decoders

Hazar Yueksel, Matthias Braendli, Andreas Burg, Giovanni Cherubini, Roy D. Cideciyan, Pier Andrea Francese, Simeon Furrer, Marcel Kossel, Lukas Kull, Danny Luu, Christian Menolfi, Thomas Morf, and Thomas Toifl

**Abstract**—The implementation of a 25.6-Gb/s four-level pulse-amplitude-modulation (4-PAM) reduced-state sliding-block Viterbi detector (VD) is presented. The power consumption of the VD is 105 mW at a supply voltage of 0.7 V, corresponding to an energy efficiency of 4.1 pJ/b. A data rate of 30.4 Gb/s is achieved with an energy efficiency of 5.3 pJ/b at a supply voltage of 0.8 V. The VD, implemented in an experimental chip fabricated in 14-nm CMOS FINFET, exploits set-partitioning principles and embedded per-survivor decision feedback to reduce implementation complexity and power consumption. The active area of the VD with 12 slices, each operating at one-eighth of the modulation rate, is  $0.507 \times 0.717 \text{ mm}^2$ . Experimental results showing system performance are obtained by using a  $(2^{15}-1)$ -bit pseudo-random binary sequence. The impact of the synchronization length and survivor path memory length on the detector design and system performance are shown. A new pipelined reduced-state sequence detector algorithm is presented for high-speed implementations. A novel speculative symbol timing recovery scheme is proposed. New simulation results are obtained to compare the performance of the Reed–Solomon (RS)-encoded 4-PAM scheme with that of the concatenated RS four-dimensional (4-D) 5-PAM trellis-coded-modulation (TCM) scheme over an ideal band-limited additive-white-Gaussian-noise channel. Drawing on the results achieved for the VD, novel design techniques for a high-speed low-complexity eight-state 4-D 5-PAM TCM decoder are proposed.

**Index Terms**—IEEE 802.3bj, IEEE 802.3bs, Viterbi detector, TCM decoder, 4-PAM, 5-PAM, four-dimensional, set partitioning, per-survivor decision feedback.

## I. INTRODUCTION

THE Viterbi algorithm (VA), first proposed in 1967 to prove an upper bound on the error probability of decoding convolutional codes [1], was later recognized as an attractive solution for symbol detection in the presence of intersymbol interference (ISI) and noise [2]. The VA minimizes the error probability in detecting the whole symbol sequence, instead of a single symbol as in decision-feedback equalization. The VA does not cancel the ISI, but rather uses

the information embedded therein to maximize the reliability of its symbol decisions.

While a maximum-likelihood sequence detector (MLSD) realizing the VA is widely used for low-symbol-rate communication systems [3], [4], the implementation complexity of an MLSD may be prohibitive for serial links operating at gigaBaud symbol rates because design specifications regarding area, latency, power consumption, and speed may not be satisfied concurrently. Instead, suboptimal solutions, such as feed-forward equalizers and decision-feedback equalizers [5], are often chosen for high-speed serial links because of their lower implementation complexity. A reduced-state sequence detector (RSSD) [6]–[8] reduces the implementation complexity of the MLSD with negligible performance degradation by exploiting set-partitioning principles [9] and embedded per-survivor decision feedback. The RSSD may also be viewed as a general form of delayed decision-feedback sequence detection [10], which uses embedded per-survivor decision feedback without set partitioning.

A sliding-block Viterbi decoder [11] breaks the speed bottleneck in sequence-decoder implementations by applying the VA to a sliding-block decoder. Compared with the sliding-block solution, a systolic-array solution [12] formulates the nonlinear add–compare–select loop of the VA as a linear vector recursion to achieve continuous parallelized operation of the VA, thereby avoiding the overhead caused by the synchronization length used for block initialization. The result of both of these works is that a sequence detector or decoder can be implemented at any speed provided that the required area, latency, and power consumption can be accommodated. The design challenge then lies in satisfying concurrently the specifications regarding area, latency, and power consumption at the target speed.

Ratified in 2014, the IEEE 802.3bj standard for 100-Gb/s Ethernet allows both two-level pulse-amplitude-modulation (2-PAM) and 4-PAM operation while targeting a data rate of 25 Gb/s per lane [13], [14]. A bit-error rate (BER)  $< 10^{-12}$  is required, which can be satisfied with a BER  $< 10^{-5}$  at the detector output if Reed–Solomon (RS) forward error correction coding is used. The emerging IEEE P802.3bs standard for 400-Gb/s Ethernet also enables both 2-PAM and 4-PAM operation while targeting a data rate of 50 Gb/s per lane for chip-to-chip and chip-to-module applications. Moreover, the IEEE 802.3ab standard for 1-Gb/s Ethernet [15] adopts four-dimensional (4-D) 5-PAM trellis-coded modulation (TCM), which uses signal-constellation expansion in conjunction with

H. Yueksel was with IBM Research – Zurich, Switzerland, EPFL, Lausanne, Switzerland, and Columbia University, NY, USA and is now with the IBM T. J. Watson Research Center, NY, USA (email: hazar.yueksel@ibm.com).

M. Braendli, G. Cherubini, R. D. Cideciyan, P. A. Francese, S. Furrer, M. Kossel, L. Kull, C. Menolfi, T. Morf, and T. Toifl are with IBM Research – Zurich, Switzerland (email: {mbr,cbi,cid,pfr,sfu,mko,lku,cme,tmr,tto}@zurich.ibm.com).

A. Burg is with EPFL, Lausanne, Switzerland (email: andreas.burg@epfl.ch).

D. Luu is with IBM Research – Zurich, Switzerland and ETH Zurich, Switzerland.

set partitioning to perform modulation and coding jointly, thus achieving coding gains for improved system robustness [9].

In this paper, we present novel algorithms and architectures for high-speed data transmission targeting future Ethernet transmission. We introduce a new pipelined reduced-state sliding-block Viterbi detector (VD) with an architecture designed to satisfy concurrently the desired design specifications for high-speed serial chip-to-chip links. The new algorithm computes partial path metrics by reducing to one the number of additions in the longest path. Furthermore, a novel speculative symbol timing recovery scheme is proposed to reduce the latency in the timing-recovery loop. We also consider a concatenated RS coding and 4-D 5-PAM TCM scheme that has the potential for being used in a future Ethernet standard. A novel TCM decoder architecture for high-speed implementations of the considered coding scheme is presented, based on a novel branch-metric unit and a novel path-metric unit. The branch-metric unit does not require as many comparators as in conventional designs, whereas the path-metric unit accounts for a pipeline stage to reduce the propagation delay of the longest path.

The rest of the paper is organized as follows. In Section II, the system model and reduced-state detection algorithms are presented. The implementation of a high-speed 4-PAM reduced-state VD is discussed in Section III, where a low-latency timing-recovery method using tentative decisions from the VD is also introduced. A new high-speed eight-state 4-D 5-PAM TCM decoder design is proposed in Section IV. The paper concludes with Section V.

## II. SYSTEM MODEL AND DETECTION ALGORITHMS

We consider an RS code that operates over the Galois Field with  $2^{10}$  field elements, where each field element corresponds to a 10-bit code symbol [13], [14]. A systematic RS encoder processes 514 message symbols to generate 14 parity symbols; i.e., the RS codewords have a length of 528 code symbols. We denote the RS code by RS(528, 514). An information symbol  $u_k$  at time  $k \in \{0, 1, \dots, K-1\}$  is drawn from an  $M$ -PAM signal constellation  $\mathbb{A}$  containing  $M \geq 2$  equidistant symbols centered around the origin; i.e.,  $\mathbb{A} = (d_0/2)\{-M+1, -M+3, \dots, M-1\}$ , where  $d_0$  is the minimum distance between symbols. A sequence of independent information symbols  $u_k$  is transmitted over a time-dispersive channel with the discrete-time impulse-response sequence  $\vec{h} = (h_0, h_1, \dots, h_\nu)$  that consists of the main-cursor coefficient  $h_0$  and post-cursor coefficients  $(h_1, h_2, \dots, h_\nu)$ . We do not include pre-cursor coefficients in the channel model because they can be canceled by a feed-forward equalizer. The channel time-dispersion length  $\nu+1$  indicates the number of symbols that the ISI spans; i.e., there are  $\nu$  neighboring symbols interfering with the transmission of a symbol over the channel. The channel is *ideal* when the number of interfering symbols  $\nu$  is 0. The ISI-channel output  $y_k = \sum_{i=0}^{\nu} h_i u_{k-i}$  is corrupted by additive white Gaussian noise (AWGN):  $z_k = y_k + w_k$ , where  $z_k$  is the detector input, and  $w_k$  is AWGN with zero mean and variance  $\sigma_w^2$ . The signal-to-noise ratio (SNR) of the channel equals  $(E_s E_h^2) / \sigma_w^2$ , where  $E_s = E\{u_k^2\}$  is the average input-symbol energy, and  $E_h^2 = \sum_{i=0}^{\nu} h_i^2$  is the

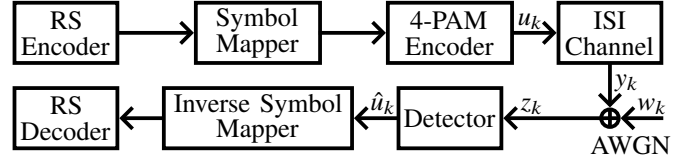


Fig. 1. 4-PAM transmission system with an RS(528, 514) code.

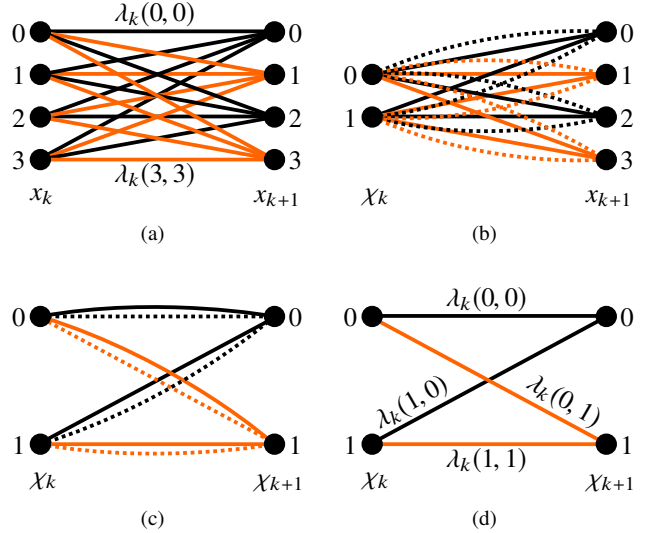


Fig. 2. Four-state trellis diagram without set partitioning (a) and with set partitioning at time  $k$  (b). Two-substate subset trellis diagram with unresolved parallel transitions (c) and resolved transitions (d).

channel-response energy. The channel state is defined as an  $M$ -ary  $\nu$ -tuple:  $x_k = (u_{k-1}, \dots, u_{k-\nu})$ . The number of channel states thus equals  $M^\nu$ .

### A. 4-PAM Transmission System

The model of the 4-PAM transmission system with an RS(528, 514) code is shown in Fig. 1. A symbol mapper and 4-PAM encoder map the output RS-encoded bits to 4-PAM symbols  $u_k$  drawn from the signal constellation  $\mathbb{A} = \{-3, -1, 1, 3\}$ . The transmission of an information symbol  $u_k$  is distorted by ISI  $\sum_{i=1}^{\nu} h_i u_{k-i}$  and corrupted by noise  $w_k$ , resulting in the input signal  $z_k$  to a detector. To reduce implementation complexity, the detector exploits embedded per-survivor decision feedback and set-partitioning principles. The channel state is therefore represented by a substate  $\chi_k$  in a reduced-state subset trellis constructed by set-partitioning principles. We partition the signal constellation  $\mathbb{A}$  into two subsets  $\mathbb{A}_0 = \{-3, 1\}$  and  $\mathbb{A}_1 = \{-1, 3\}$  such that the minimum intrasubset Euclidean distance is maximized.

As an example, the full-state trellis diagram for a channel time-dispersion length of 2 is shown in Fig. 2a, where the branch metrics quantifying the likelihood of a transition from a state  $x_k$  to a state  $x_{k+1}$  are denoted by  $\lambda_k(x_k, x_{k+1})$ . When we join the states  $x_k \in \{0, 2\}$  in Fig. 2a, we obtain the substate  $\chi_k = 0$  in Fig. 2b. Similarly, when we join the states  $x_k \in \{1, 3\}$  in Fig. 2a, we obtain the substate  $\chi_k = 1$  in Fig. 2b, where the dashed lines represent parallel transitions. When the same procedure is executed at time  $k+1$ , we obtain

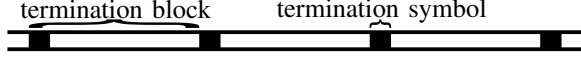


Fig. 3. Transmission scheme of the IEEE 802.3bj standard with termination blocks and termination symbols [16].

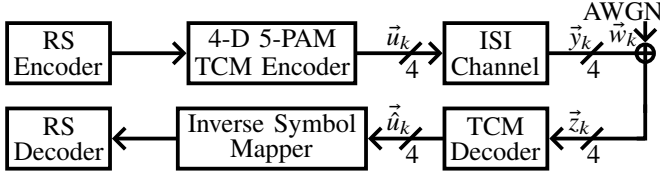


Fig. 4. Concatenated outer RS(528, 514) code and inner 4-D 5-PAM TCM transmission system.

the trellis diagram depicted in Fig. 2c. Once all the parallel transitions have been resolved by finding the transitions with the minimum branch metric, the trellis diagram in Fig. 2d is obtained.

The branch metrics  $\lambda_k(\chi_k, \chi_{k+1})$  associated with the reduced-state subset trellis shown in Fig. 2d equal  $(z_k - \sum_{i=0}^n h_i u_{k-i} - \sum_{i=n+1}^v h_i \hat{u}_{k-i})^2$ , where  $n \in \{1, 2, \dots, v\}$ , and  $\hat{u}_k$  denotes a tentative symbol decision. The third term  $\sum_{i=n+1}^v h_i \hat{u}_{k-i}$  represents embedded per-survivor decision feedback with  $v - n$  taps. As the branch metric, the squared Euclidean distance (SED) can be replaced with the Euclidean distance (ED) to reduce implementation complexity because the resulting performance degradation is negligible [17].

The IEEE 802.3bj standard [13], [14] specifies a transmission scheme that adopts state pinning as depicted in Fig. 3. A *termination block* is defined as a block of symbols that starts with and is followed by a pseudo-random symbol referred to as a *termination symbol* as depicted in Fig. 3. The pseudo-random termination symbols can be generated at the receiver after receiving a training sequence. The survivor paths merge without ambiguity into the terminal symbol initiating a termination block when the channel time-dispersion length is 2. If the channel time-dispersion length is greater than 2, then an overhead symbol sequence for block initialization must be processed for achieving block independence when a sliding-block detector is used. Regardless of the channel time-dispersion length, the survivor paths obtained in the detector merge without ambiguity into the termination symbol following a termination block. Therefore, it is then not necessary to process an overhead symbol sequence for block termination in order to achieve block independence.

### B. 4-D 5-PAM TCM Transmission System

The model of the 4-D 5-PAM TCM transmission system with an outer RS(528, 514) code is shown in Fig. 4. The RS encoder outputs bits to the 4-D 5-PAM TCM encoder which encodes eight incoming bits  $(b_0, b_1, \dots, b_7)$  to nine bits  $(b_0, b_1, \dots, b_8)$  by a convolutional encoder as shown in Fig. 5a [15]. The radix-4 trellis diagram with eight states corresponding to the three-bit state of the convolutional encoder is shown in Fig. 5b. The signal constellation  $\mathbb{A} = \{-2, -1, 0, 1, 2\}$  does not consist of equiprobable symbols, resulting in an average

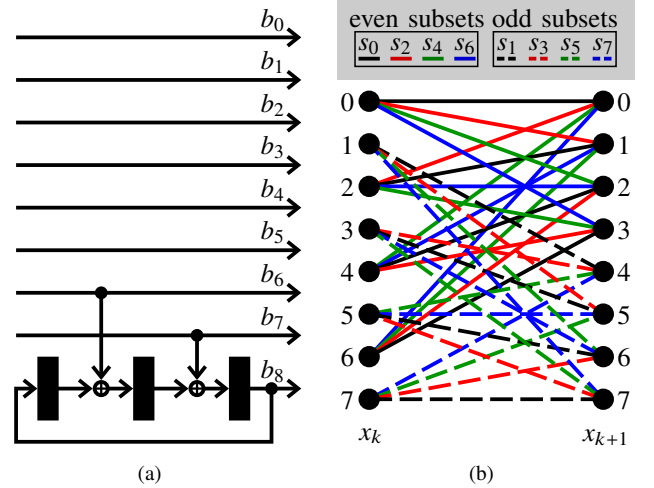


Fig. 5. 2/3-rate convolutional encoder (a) and eight-state trellis diagram (b) of the 4-D 5-PAM TCM scheme with even and odd 4-D subsets.

input-symbol energy  $E_s$  of 1.8125 corresponding to a shaping gain of  $|10 \log(1.8125/2)| \approx 0.4$  dB.

Similarly to the 4-PAM transmission system, the signal constellation  $\mathbb{A}$  is partitioned into two subsets  $\mathbb{A}_0 = \{-2, 0, 2\}$  and  $\mathbb{A}_1 = \{-1, 1\}$  such that the minimum intrasubset Euclidean distance is maximized. This results in 16 different 4-D subsets  $\{(\mathbb{A}_0, \mathbb{A}_0, \mathbb{A}_0, \mathbb{A}_0), (\mathbb{A}_0, \mathbb{A}_0, \mathbb{A}_0, \mathbb{A}_1), \dots, (\mathbb{A}_1, \mathbb{A}_1, \mathbb{A}_1, \mathbb{A}_1)\}$ . By uniting a 4-D subset and its complement; e.g.,  $(\mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_0, \mathbb{A}_1) = (\mathbb{A}_1, \mathbb{A}_0, \mathbb{A}_1, \mathbb{A}_0)$ , eight new 4-D subsets  $\{s_0, s_1, \dots, s_7\}$  are obtained such that the minimum 4-D intrasubset Euclidean distance remains constant. It can readily be shown that the cardinality  $\{|s_0|, |s_1|, \dots, |s_7|\}$  of all 4-D subsets is at least 64, which corresponds to the number of points used in a 4-D subset  $\{s_0, s_1, \dots, s_7\}$ . The bits  $(b_6, b_7, b_8)$  are used to select one of the eight 4-D subsets  $\{s_0, s_1, \dots, s_7\}$ , and the remaining 6 bits  $(b_0, b_1, \dots, b_5)$  are used to select a 4-D symbol in that 4-D subset. The inverse symbol mapper performs symbol-to-bit mapping via the tables given in [15]. Compared with the average error probability per symbol of uncoded 5-PAM for an ideal band-limited AWGN channel, an asymptotic distance gain of 6 dB is achieved.

### C. BER-Performance Comparison

The BER performance achieved by the RS(528, 514)-encoded 4-PAM, the 4-D 5-PAM TCM, and the concatenated outer RS(528, 514) code and 4-D 5-PAM inner TCM over an ideal band-limited AWGN channel is shown in Fig. 6. The average uncoded 4-PAM bit-error probability  $P(4, \epsilon)$  for an ideal band-limited AWGN channel assuming Gray mapping and the average uncoded 5-PAM bit-error probability  $P(5, \epsilon)$  for an ideal band-limited AWGN channel assuming one bit error per symbol error are shown in dashed and dash-dotted lines, respectively. The RS(528, 514) code used in the 4-PAM transmission system achieves a simulated SNR gain of 3 dB over  $P(4, \epsilon)$  at a BER of  $10^{-6}$ . Simulation results indicate that the performance of the 4-D 5-PAM TCM coincides with that of the RS(528, 514)-encoded 4-PAM system at a BER of  $6 \times 10^{-7}$  at an SNR of 17.4 dB and yields better performance at lower SNRs. The concatenated outer RS(528, 514) code



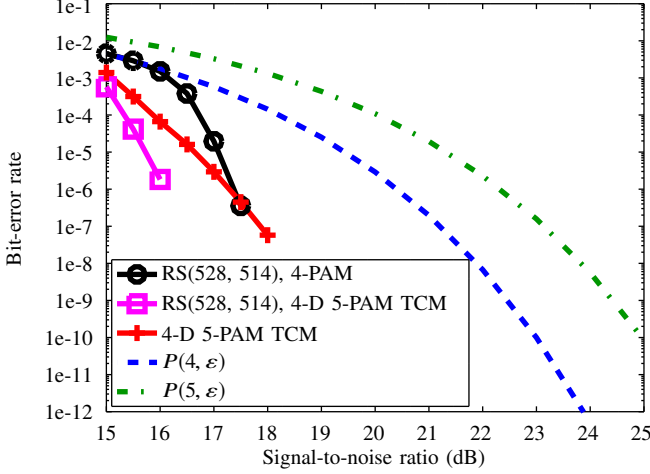


Fig. 6. BER-performance comparison between the RS(528, 514)-encoded 4-PAM and the concatenated RS(528, 514) 4-D 5-PAM TCM schemes over an ideal band-limited AWGN channel.

and 4-D 5-PAM inner TCM achieves a simulated SNR gain of approximately 4 dB over  $P(4, \epsilon)$  and 6 dB over  $P(5, \epsilon)$  at a BER of  $2 \times 10^{-6}$  without bandwidth expansion. These results are also relevant for time-dispersive channels because the ISI in such a system can be mitigated by methods such as Tomlinson–Harashima precoding [18]. We note that  $P(4, \epsilon)$  and  $P(5, \epsilon)$  are analytical expressions denoting the theoretical limit on bit-error probabilities of uncoded transmission over an AWGN channel without intersymbol interference. Therefore, they denote neither the performance of reduced-state sequence detection nor that of maximum-likelihood sequence detection because there is no need to use sequence detection for transmission over an AWGN channel when there is no intersymbol interference.  $P(4, \epsilon)$  and  $P(5, \epsilon)$  are introduced to show the performance gain of the considered transmission schemes over the ideal cases.

#### D. Viterbi Algorithm

The VA used for symbol detection in the presence of ISI and noise [2] for a radix- $M^v$  trellis is presented in Algorithm 1 to introduce our notation. The algorithm is initialized at time  $k$ , and the path metrics  $\Gamma_k(x_k)$  for all states  $x_k \in \{0, 1, \dots, M^v - 1\}$  are set to 0. The branch metrics  $\lambda_k(x_k, x_{k+1})$  are calculated in each time step  $k \in \{0, 1, \dots, K - 1\}$  as the distance of the respective hypothesized input signals  $\tilde{y}(x_k, x_{k+1})$  from the input sample  $z_k$ . The distance measure is shown in Algorithm 1 as the SED. The partial path metrics  $\Gamma_k(x_k, x_{k+1})$  are calculated as the sum of the path metrics  $\Gamma_k(x_k)$  and the corresponding branch metrics  $\lambda_k(x_k, x_{k+1})$ . For each  $x_{k+1} \in \{0, 1, \dots, M^v - 1\}$ , the updated path metric  $\Gamma_{k+1}(x_{k+1})$  is selected as the minimum of the partial path metrics  $\Gamma_k(x_k, x_{k+1})$ . The state  $x_k$  is detected as  $\hat{x}_k(x_{k+1})$ , which is the argument of the minimum of the partial path metrics  $\Gamma_k(x_k, x_{k+1})$ . The symbol decision  $\hat{u}_k(x_{k+1})$  is performed by the inverse symbol-mapper function  $f_x(\cdot)$  whose input is the detected state  $\hat{x}_k(x_{k+1})$ . Depending on the desired design specifications, a memory-organization technique, such as the traceback and register-exchange methods [19], is

---

#### Algorithm 1: VA for a radix- $M^v$ trellis.

---

**Input:**  $z_k, \tilde{y}$ ;  
**Output:**  $\hat{u}_k, \hat{x}_k$ ;  
*Initialization*  
1:  $k \leftarrow 0$ ;  
2: **for**  $x_k = 0$  to  $M^v - 1$  **do**  
3:  $\Gamma_k(x_k) \leftarrow 0$ ;  
4: **end for**  
*Branch- and path-metric computation*  
5: **for**  $k = 0$  to  $K - 1$  **do**  
6: **for**  $x_{k+1} = 0$  to  $M^v - 1$  **do**  
7: **for**  $x_k = 0$  to  $M^v - 1$  **do**  
8:  $\lambda_k(x_k, x_{k+1}) \leftarrow (z_k - \tilde{y}(x_k, x_{k+1}))^2$ ;  
9:  $\Gamma_k(x_k, x_{k+1}) \leftarrow \Gamma_k(x_k) + \lambda_k(x_k, x_{k+1})$ ;  
10: **end for**  
11:  $\Gamma_{k+1}(x_{k+1}) \leftarrow \min_{x_k} \Gamma_k(x_k, x_{k+1})$ ;  
12:  $\hat{x}_k(x_{k+1}) \leftarrow \arg \min_{x_k} \Gamma_k(x_k, x_{k+1})$ ;  
13:  $\hat{u}_k(x_{k+1}) \leftarrow f_x(\hat{x}_k(x_{k+1}))$ ;  
14: **end for**  
15: **end for**

---

adopted thereafter to store the survivor paths corresponding to each state  $x_k$ . The competing survivor paths originating from any possible initial state  $x_0$  in the trellis representing the state transitions merge with high probability after a number of iterations  $\alpha$  known as the *synchronization length*. Similarly, starting from any terminal state  $x_{K-1}$ , these survivor paths will, with high probability, merge with the true survivor sequence a number of iterations  $\beta$  back in the trellis. The parameter  $\beta$  is known as the *survivor path memory length*. The survivor sequence is selected as one of the survivor paths after a number of time steps corresponding to the survivor path memory length  $\beta$ . It was recently demonstrated that the synchronization length  $\alpha$  and survivor path memory length  $\beta$  differ significantly in an optimized VD [17].

#### E. New High-Speed RSSD Algorithm

Implementing the VA such that design specifications of high-speed serial links regarding area, latency, power consumption, and speed are satisfied often requires modifying Algorithm 1. The new high-speed RSSD algorithm for the radix-2 subset trellis shown in Fig. 2d is presented in Algorithm 2. Set partitioning is introduced such that the number of substates is 2; i.e.,  $\chi_k \in \{0, 1\}$  as shown in Fig. 2d. Embedded per-survivor decision feedback is integrated into the calculation of the hypothesized input values  $\tilde{y}_k(\hat{u}_{k-1}, u_k)$ . Computing partial path metrics  $\Gamma_k(\chi_k, \chi_{k+1})$  is delayed by one time step to avoid having two additions in the same clock cycle. The path metrics  $\Gamma_k(\chi_k)$  are also updated after the mentioned delay. Similarly to Algorithm 1, the substate  $\chi_{k-1}$  is detected as  $\hat{\chi}_{k-1}$ , which is the argument of the minimum of the partial path metrics  $\Gamma_{k-1}(\chi_{k-1}, \chi_k)$ . Once the resolved subset decisions  $\hat{\zeta}_{k-1}(\chi_k)$  have been chosen out of the unresolved subset decisions  $\hat{\zeta}_{k-1}(\chi_{k-1}, \chi_k)$ , they are used as inputs to the inverse symbol-mapper function  $f_{\chi}(\cdot)$  to make symbol decisions  $\hat{u}_{k-1}(\chi_k)$ . Similarly to Algorithm 1, a memory-organization technique would have to be used for storing the survivor paths corresponding to each substate

---

**Algorithm 2:** New high-speed RSSD algorithm.
 

---

**Input:**  $z_k, \tilde{y}_k$ ;  
**Output:**  $\hat{u}_k, \hat{\chi}_k$ ;

*Initialization*

- 1:  $k \leftarrow 0$ ;
- 2: **for**  $\chi_k = 0$  to 1 **do**
- 3:    $\Gamma_k(\chi_k) \leftarrow 0$ ;
- 4: **end for**

*Branch- and path-metric computation*

- 5: **for**  $k = 0$  to  $K - 1$  **do**
- 6:   **for**  $\chi_k = 0$  to 1 **do**
- 7:     **for**  $\chi_{k-1} = 0$  to 1 **do**
- 8:       **for**  $\chi_{k+1} = 0$  to 1 **do**
- 9:           $q_k \leftarrow |z_k - \tilde{y}_k(\mathbb{A}_{\chi_k}(\hat{\zeta}_{k-1}(\chi_k)), u_k \in \mathbb{A}_{\chi_{k+1}})|$ ;
- 10:           $\lambda_k(\chi_k, \chi_{k+1}) \leftarrow \min_{u_k} q_k$ ;
- 11:           $\hat{\zeta}_k(\chi_k, \chi_{k+1}) \leftarrow \arg \min_{u_k} q_k$ ;
- 12:       **end for**
- 13:        $\Gamma_{k-1}(\chi_{k-1}, \chi_k) \leftarrow \Gamma_{k-1}(\chi_{k-1}) + \lambda_{k-1}(\chi_{k-1}, \chi_k)$ ;
- 14:     **end for**
- 15:      $\Gamma_k(\chi_k) \leftarrow \min_{\chi_{k-1}} \Gamma_{k-1}(\chi_{k-1}, \chi_k)$ ;
- 16:      $\hat{\chi}_{k-1}(\chi_k) \leftarrow \arg \min_{\chi_{k-1}} \Gamma_{k-1}(\chi_{k-1}, \chi_k)$ ;
- 17:      $\hat{\zeta}_{k-1}(\chi_k) \leftarrow \hat{\zeta}_{k-1}(\hat{\chi}_{k-1}(\chi_k), \chi_k)$ ;
- 18:      $\hat{u}_{k-1}(\chi_k) \leftarrow f_{\chi}(\hat{\zeta}_{k-1}(\chi_k))$ ;
- 19:   **end for**
- 20: **end for**

---

$\chi_k$ . However, if termination symbols are enabled, the survivor sequence can be output as termination blocks without having to account for the survivor path memory length  $\beta$ . While the references [6]–[8] introduce state reduction to the VA, a novel method to pipeline the VA with state reduction for high-speed implementations is proposed in Algorithm 2.

### III. HIGH-SPEED 4-PAM VD IMPLEMENTATION

The proposed VD is implemented in an experimental chip fabricated in 14-nm twin-well CMOS FINFET technology on p-silicon-on-insulator substrate. 15 metal layers are available in the process and used in the circuit. The whole circuit is designed in VHDL and synthesized. Its physical design uses only standard logic cells.

#### A. System Architecture

The system architecture of our test chip is shown in Fig. 7, which includes a pseudo-random binary sequence (PRBS) generator, channel emulator, eighth-rate VD, and PRBS checker. All logic circuits receive an eighth-rate (c8) clock signal. The PRBS generator outputs 16 input bits in the  $k^{\text{th}}$  c8 clock period, which are mapped to eight input symbols  $\{u_{k,1}, u_{k,2}, \dots, u_{k,8}\}$  drawn from the 4-PAM signal constellation  $\mathbb{A}$ . A substate  $\chi_k$  in the reduced-state subset trellis shown in Fig. 2d is defined as the subset  $\zeta_{k-1} \in \{\mathbb{A}_0, \mathbb{A}_1\}$  to which the information symbol  $u_{k-1}$  belongs. For example, when the information symbol  $u_{k-1}$  is in the subset  $\mathbb{A}_0$ , we represent the substate  $\chi_k$  by 0. Similarly, when the information symbol  $u_{k-1}$  is in the subset  $\mathbb{A}_1$ , we represent the substate  $\chi_k$  by 1. By the same notation, an information symbol  $u_k$  is represented by its index in the signal constellation  $\mathbb{A}$ :  $u_k \in \{0, 1, 2, 3\}$ . For example, when the information symbol  $u_{k-1}$  is  $-3$ , then we represent the information symbol  $u_{k-1}$  by 0. The PRBS

generator also outputs the termination substate  $\chi_{k-l, \text{term}}$ , which represents the substate to which the information symbol  $u_{k,8}$  belongs, where  $l$  is the latency from the output of the PRBS generator to the input of the RSSDs.

A channel emulator convolves the sequence of symbols  $\{u_k\}$  with the assumed discrete-time impulse-response sequence  $\vec{h} = (h_0, h_1, h_2, h_3)$ , where, without loss of generality, the main-cursor channel coefficient  $h_0$  is normalized to 1. A reduced-rate implementation of the RSSD is needed to reach the target data rates due to electromigration concerns and the limitations on the maximum clock frequency of the 14-nm CMOS technology and its digital-design flow. An eighth-rate implementation is chosen because its higher rate counterparts could not reach the target data rates, and its lower rate counterparts would introduce more latency and occupy more area. The eighth-rate VD consists of a synchronizer, a serial-to-block converter, parallel RSSDs, a register array, and a block-to-serial converter. The synchronizer is a state machine that organizes the data flow within the VD. It takes as inputs the termination substate  $\chi_{k-l, \text{term}}$  to output it after  $l$  c8 clock periods and the signal “mode” that determines the synchronization length  $\alpha \in \{0, 24\}$  for initialization and the number of RSSDs operating in parallel. When the synchronization length  $\alpha$  is 0, then the number of RSSDs operating in parallel is eight because the VD operates at one-eighth of the modulation rate. When the synchronization length  $\alpha$  is 24, then the number of RSSDs operating in parallel is 12 due to the resulting 50% overhead of information symbols, as the termination block length is 48 symbols. The serial-to-block converter, controlled by the signal “flag\_vmux”, distributes the “serial” channel-output signals to the RSSDs. It receives eight input signals per c8 clock period, but outputs only one of them to an RSSD. The block-to-serial converter, controlled by the signal “flag\_vdemux”, reorganizes the termination blocks such that they can be output “serially” to the PRBS checker used as the error counter. The register array stores the discrete-time impulse-response sequence  $\vec{h}$ , pre-computed partial hypothesized input values  $\vec{y}'_k$ , and the signal “mode”.

#### B. Reduced-State Sequence Detector

The RSSD shown in Fig. 8 has two substates and two embedded post-cursor per-survivor decision-feedback taps for  $\{h_2, h_3\}$ . The RSSD consists of a hypothesized value generator (HVG), a branch metric unit (BMU), a path metric unit (PMU), and a survivor memory unit (SMU). The sequence  $\vec{h}$ , the previous substate decisions  $\vec{\chi}_{k-1} = (\hat{\chi}_{k-1}(0), \hat{\chi}_{k-1}(1))$ , and tentative symbols decisions  $\vec{\hat{u}}_{k-1} = (\hat{u}_{k-1}(0), \hat{u}_{k-1}(1))$  are sent to the HVG to compute the hypothesized input values  $\vec{y}'_k(\hat{u}_{k-1}, u_k)$  with which to compare the input signal  $y_k$  in the BMU. The parallel outcomes of these comparisons are used by the BMU to calculate the branch metrics  $\vec{\lambda}_k(\chi_k, \chi_{k+1})$  and unresolved subset decisions  $\vec{\zeta}_k(\chi_k, \chi_{k+1})$ , which are then sent to the PMU to make symbol decisions  $\vec{\hat{u}}_{k-1}$ , substate decisions  $\vec{\chi}_{k-1}$ , and resolved subset decisions  $\vec{\zeta}_{k-1}$ . The symbol decisions  $\vec{\hat{u}}_{k-1}$  and the substate decisions  $\vec{\chi}_{k-1}$  are used by the SMU for storing survivor paths from which the detected survivor sequence is retrieved. Metric precisions are chosen as described in [11].



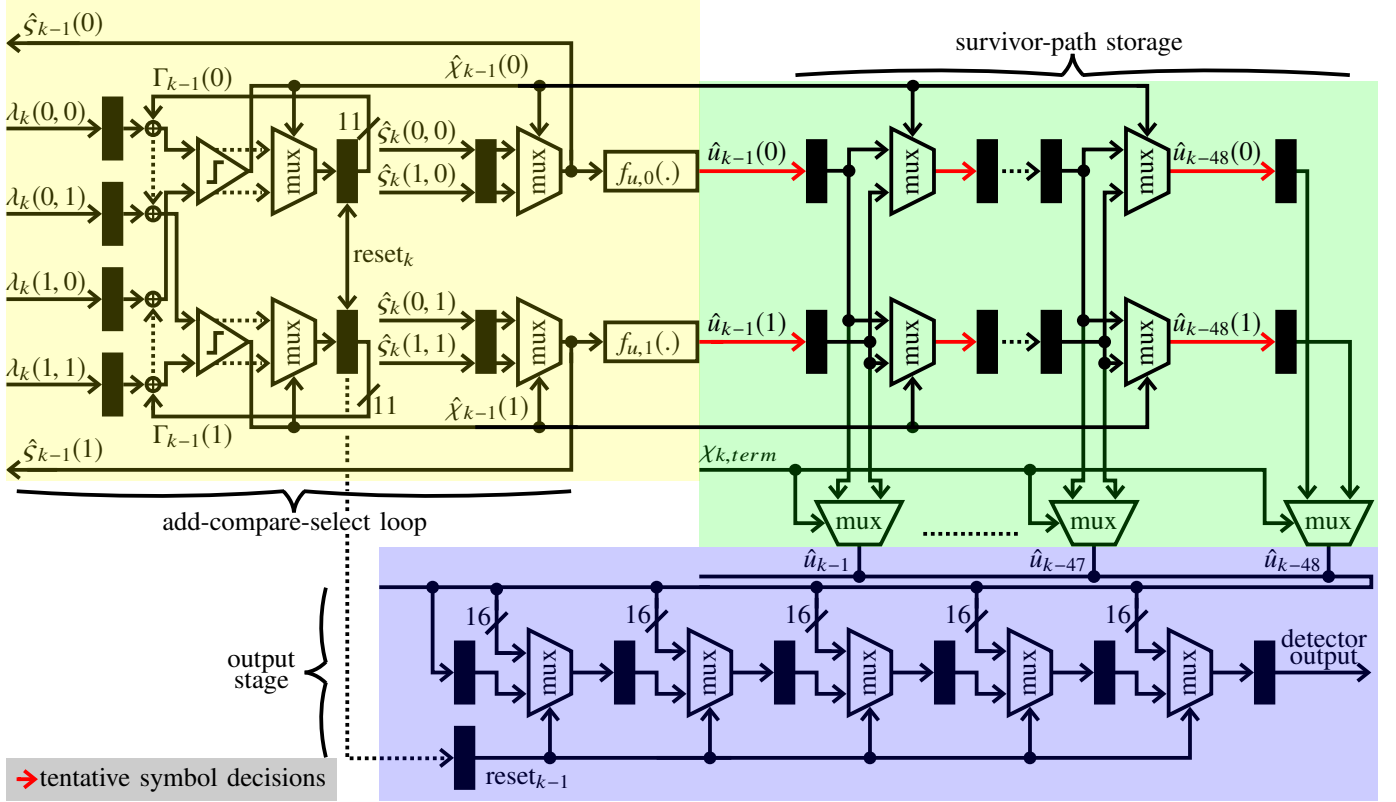


Fig. 11. Block diagram of the PMU and SMU.

2) *Branch Metric Unit*: The BMU shown in Fig. 10 computes the branch metrics  $\vec{\lambda}_k(\chi_k, \chi_{k+1})$ , which represent the likelihood of a transition in the trellis diagram shown in Fig. 2d. The smaller the branch metric  $\lambda_k(\chi_k, \chi_{k+1})$ , the more likely the transition from substate  $\chi_k$  to substate  $\chi_{k+1}$  represented by that branch metric  $\lambda_k(\chi_k, \chi_{k+1})$ . To reduce hardware complexity, the ED is chosen instead of the SED as the branch metric because the resulting performance degradation is negligible [17]. The EDs  $\vec{d}_k(\hat{u}_{k-1}, u_k) = |y_k - \bar{y}_k(\hat{u}_{k-1}, u_k)|$  are used to calculate the branch metric of the parallel transitions  $\vec{p}_k(\chi_k, u_k) = \vec{d}_k(\hat{u}_{k-1}, u_k)$ ,  $\forall u_k \in \mathbb{A}$ ,  $\forall \hat{u}_{k-1} \in \hat{\zeta}_{k-1}$ . The resolved branch metrics are then obtained:  $\vec{\lambda}_k(\chi_k, \chi_{k+1}) = \min_{u_k \in \zeta_k} \vec{p}_k(\chi_k, u_k)$ . The unresolved subset decisions  $\vec{\zeta}_k(\chi_k, \chi_{k+1})$  that determine which parallel transitions have survived are output to the PMU for making tentative symbol decisions  $\{\hat{u}_{k-1}(0), \hat{u}_{k-1}(1)\}$ . The previously resolved subset decisions  $\{\hat{\zeta}_{k-1}(0), \hat{\zeta}_{k-1}(1)\}$  incoming from the PMU are used to resolve parallel transitions in the BMU at time  $k$ .

3) *Path Metric Unit*: The PMU highlighted in yellow in Fig. 11 computes the path metrics  $\vec{\Gamma}_k(\chi_k)$  to make substate decisions  $\hat{\chi}_k$  and symbol decisions  $\hat{u}_k$ . The path metrics  $\vec{\Gamma}_k(\chi_k)$  are the accumulation of branch metrics of a path in the trellis. The substate is detected from the comparison between its respective partial path metrics as  $\hat{\chi}_k = \arg \min_{\chi_k} \vec{\Gamma}_k(\chi_k, \chi_{k+1})$ . One of the partial path metrics  $\vec{\Gamma}_k(\chi_k, \chi_{k+1}) = \vec{\Gamma}_k(\chi_k) + \lambda_k(\chi_k, \chi_{k+1})$  becomes the respective path metric depending on the detected substate  $\hat{\chi}_{k-1}(\chi_k)$ :  $\vec{\Gamma}_{k+1}(\chi_{k+1}) = \min_{\chi_k} \vec{\Gamma}_k(\chi_k, \chi_{k+1})$ . The detected substate  $\hat{\chi}_k$

also chooses its respective resolved subset decision  $\hat{\zeta}_k(\chi_k)$  out of the unresolved subset decisions  $\vec{\zeta}_k(\chi_k, \chi_{k+1})$ . The resolved subset decision  $\hat{\zeta}_k(\chi_k) = \min_{u_k \in \zeta_k} \zeta_k(\chi_k, \chi_{k+1})$  is then mapped to a symbol decision  $\hat{u}_k(\chi_k)$  by the symbol-mapper functions that bear the set-partitioning information  $f_{u,0}(\cdot)$  and  $f_{u,1}(\cdot)$ :  $f_{u,0}(\hat{\zeta}_{k-1}(0)) = \hat{u}_{k-1}(0)$ ;  $f_{u,1}(\hat{\zeta}_{k-1}(1)) = \hat{u}_{k-1}(1)$ . The branch metrics and corresponding path metrics are calculated in subsequent c8 clock periods, thus reducing the number of additions in the longest path to one, when  $h_2 = h_3 = 0$ . The “reset<sub>k</sub>” signal resets the path metrics at the end of a termination block so that the incoming termination block can be processed without bias from the current termination block.

4) *Survivor Memory Unit*: The SMU is highlighted in green and blue in Fig. 11. The register-exchange method [19] is used as the memory-organization technique, and the traceback method could as well be used in this architecture when  $h_2 \neq 0$  and  $h_3 \neq 0$ . The termination substate  $\chi_{k,\text{term}}$  determines which of the two competing survivor paths succeeds in being the survivor sequence. The green-highlighted circuitry stores the survivor paths, whereas the blue-highlighted circuitry chooses and outputs the survivor sequence. The “reset<sub>k-1</sub>” signal is asserted when the last termination symbol of a termination block is processed for the survivor sequence to be retrieved from the SMU. The detector then outputs eight symbols, which correspond to 16 bits, per c8 clock period to match the speed of the PRBS checker.

### C. Experimental Results

As the focus of this work is to demonstrate a high-speed sequence detector, a sequence of 4-PAM encoded symbols

TABLE I  
PERFORMANCE COMPARISON FOR HIGH-SPEED VDS

	[21]	[22]	[23]	[24]	[25]	This work
Technology	250-nm CMOS	130-nm CMOS	90-nm CMOS	180-nm SiGe BiCMOS	90-nm CMOS	14-nm CMOS
Data rate (Gb/s)	0.2	2.8	1.9	11.1	32	25.6
Supply (V)	2.5	1.5	1.7	{3.3, 1.8, -1.2}	1	0.7
Power (mW)	55	2200	358	2520	2390	105
Energy efficiency (pJ/b)	275	785.7	188.4	227	74.7	4.1
Modulation	4-PAM	2-PAM	2-PAM	2-PAM	2-PAM	4-PAM
Design type	Analog	Digital	Digital	Analog	Digital	Digital

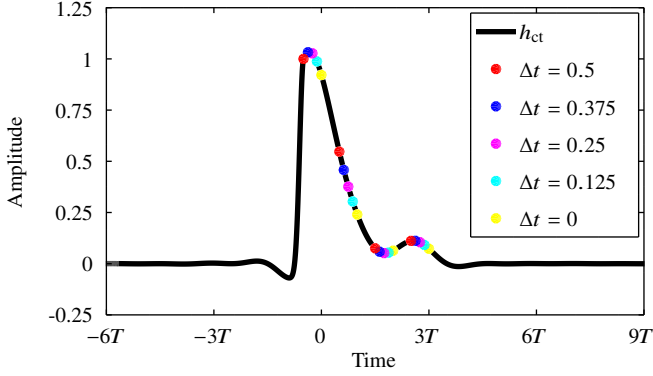


Fig. 12. Channel impulse response and sampling points (modified from [16]).

was generated by a  $(2^{15}-1)$ -bit PRBS and transmitted over the emulated channel. A practical channel model is taken from the website [20] of the IEEE P802.3bs standard and is equalized by using the channel operating margin program [14]. The resulting continuous-time channel impulse response  $h_{ct}$  is shown in Fig. 12, where the yellow points correspond to the sampling instants for which the channel-response energy is minimal. The system BER is measured with different channel models obtained by sampling the continuous-time channel impulse response  $h_{ct}$  with different normalized time offsets  $\Delta t \triangleq |t - nT|/T$  associated with the channel coefficient  $h_n$ , where  $t$  is time, and  $T$  is the modulation interval.

A plot showing measured power consumption versus the clock frequency of the VD is given in Fig. 13 for the functional operation points at the two supply voltages  $\{0.8\text{ V}, 0.7\text{ V}\}$  used in the measurements. The maximum operating clock frequency is 1.9 GHz, which corresponds to a maximum data rate of 30.4 Gb/s, as an eighth-rate 4-PAM VD is used. At a supply of 0.8 V or 0.7 V, a data rate of 30.4 Gb/s or 25.6 Gb/s is achieved with an energy efficiency of 5.29 pJ/b or 4.09 pJ/b, respectively. For hardware verification, the BER performance obtained from measurements and simulations for the synchronization length  $\alpha$  equal to 0 is shown in Fig. 14. For the synchronization length  $\alpha$  equal to 24, error-free operation is obtained for the normalized time offset  $\Delta t \in \{0, 0.125, 0.25, 0.375, 0.5\}$ . At a data rate of 25.6 Gb/s, the latency of an RSSD is 31 ns and 46 ns for  $\alpha = 0$  and  $\alpha = 24$ , respectively. The chip layout and micrograph are shown in Fig. 15a and Fig. 15b, respectively, where major circuit blocks are highlighted according to the chip

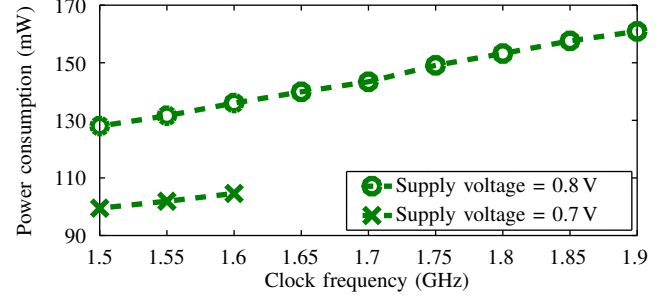


Fig. 13. Power consumption versus clock frequency [16].

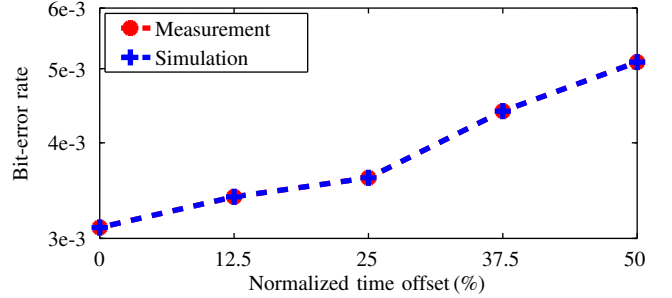


Fig. 14. BER performance of the VD when the synchronization length  $\alpha$  is equal to 0 (modified from [16]).

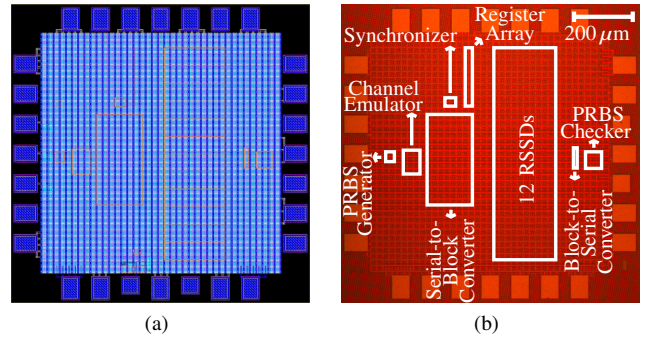


Fig. 15. Chip layout (a) and micrograph (b) [16].

floorplan. The total chip area is  $1 \times 1\text{ mm}^2$ . The 12 RSSDs occupy an area of  $0.212 \times 0.708\text{ mm}^2$ , each of which has 85917 transistors. The synchronizer, serial-to-block converter, block-to-serial converter, and register arrays occupy an area of  $34 \times 31\ \mu\text{m}^2$ ,  $153 \times 308\ \mu\text{m}^2$ ,  $14 \times 68\ \mu\text{m}^2$ , and  $20 \times 191\ \mu\text{m}^2$ , respectively. The total synthesized VD area is  $507 \times 717\ \mu\text{m}^2$ . Table I summarizes the VD performance compared with



that of other reported high-speed VDs. When comparing the performance of VDs, because of their ability of attaining block independence, a fair method of comparison is to evaluate their energy efficiency and area use at the same data rate. According to the International Technology Roadmap for Semiconductors (ITRS) [26]–[28], a technology scaling factor of 5.3 can be derived between the 90-nm and 14-nm technologies. Dividing the power consumption of [25] by 2 to normalize to 4-PAM and accounting for the above scaling factor, our work has a 34% better energy efficiency than [25]. According to the transistor density figures from the ITRS reports [26]–[28], the scaling factor between the 90-nm and 14-nm technologies is 40.4. Our work uses 4-PAM whereas [25] uses 2-PAM, and our work accounts for 3 interfering channel coefficients whereas [25] accounts for 2 interfering channel coefficients. This means that our work deals with 64 channel states whereas [25] deals with 4 channel states. Therefore, adjusting for the difference in architecture between [25] and our work, our work's data-rate-per-area figure is 114% better than [25].

#### D. Speculative Symbol Timing Recovery

High-speed receivers need pipelining in order to meet the data-rate requirements, which increases latency. Because of the increasing data-rate requirements, a reliable timing-recovery scheme with low latency is crucial for high-speed receiver design. DFEs' error-rate performance may not suffice for timing-recovery purposes in high-speed implementations, and precoding is required to counteract the error propagation incurred by using a DFE. We propose a method for timing recovery based on using multiple loop filters and speculative tentative symbol decisions. The number of latency  $l_{SMU}$  cycles of the SMU equals at least the survivor path memory length  $\beta$  when there is no termination-block scheme in place:  $l \geq \beta t_{clk}$ , where  $t_{clk}$  is the clock period. The number of latency  $l_{SMU}$  cycles of the SMU equals at least the termination-block length  $\rho$  when there is a termination-block scheme in place:  $l \geq \rho t_{clk}$  [17].

In the minimal embodiment with two substates, because the symbol decisions at the output of the PMU are provided to the timing-error detectors, a select signal  $s_k$  from the PMU should be output to the multiplexers following the register placed after the loop filter. In the minimal embodiment, this select signal determines which of the phases and gradients are selected based on the path metrics, not on the candidate path metrics. In an implementation wherein a greater path metric means a less likely path, the select signal is  $s_k = \hat{\chi}_k(\arg\min_{\chi_k} \Gamma_k(\chi_k))$  when there is no register stage within the loop filter.

The error signal

$$\Delta\tau_{k-1} = -z_{k-1}\hat{u}_{k-2} + z_{k-2}\hat{u}_{k-1}. \quad (1)$$

is used in the equations describing the operation of the loop filter

$$\tau_k = \tau_{k-1} + \gamma\Delta\tau_{k-1} + \Delta T_{k-1} + \zeta\Delta\tau_{k-1} \quad (2)$$

$$\Delta T_k = \Delta T_{k-1} + \zeta\Delta\tau_{k-1}, \quad (3)$$

where  $\tau$  denotes the timing phase,  $\gamma$  and  $\zeta$  denote the loop gains, and  $\Delta T$  denotes the frequency offset compensating the difference between the transmitter and receiver clocks [29].

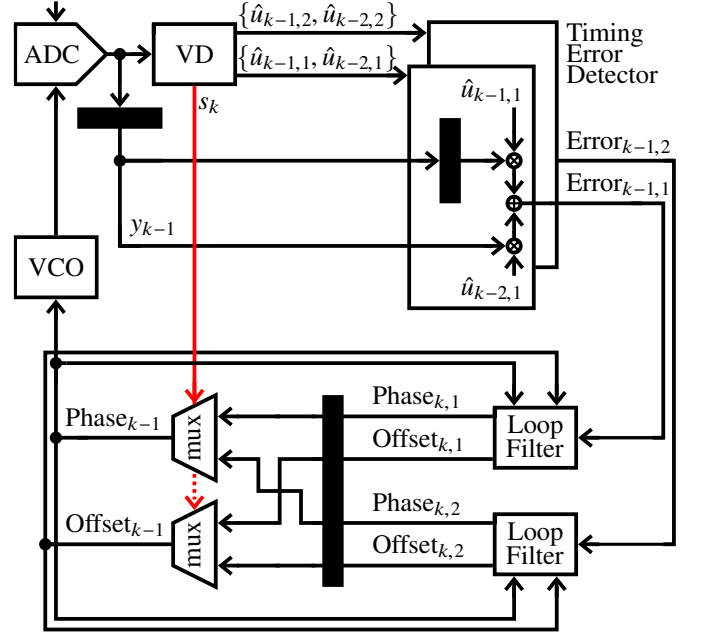


Fig. 16. Block diagram of the speculative symbol timing recovery scheme.

TABLE II  
SELECT SIGNAL: ONE REGISTER STAGE WITHIN LOOP FILTER

$\hat{\chi}_k(0)$	$\hat{\chi}_k(1)$	$\hat{\chi}_{k+1}(0)$	$\hat{\chi}_{k+1}(1)$	$s_{k+1}$ if ( $\Theta$ )	$s_{k+1}$ if ( $\neg\Theta$ )
0	0	X	X	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	1	0	0
1	1	X	X	1	1

Accordingly, the speculative error signals

$$\Delta\tau_{k-1,1} = -z_{k-1}\hat{u}_{k-2,1} + z_{k-2}\hat{u}_{k-1,1} \quad (4)$$

$$\Delta\tau_{k-1,2} = -z_{k-1}\hat{u}_{k-2,2} + z_{k-2}\hat{u}_{k-1,2} \quad (5)$$

are used to calculate the timing phase and frequency offset signals as follows.

$$\tau_{k,1} = \tau_{k-1} + \gamma\Delta\tau_{k-1,1} + \Delta T_{k-1} + \zeta\Delta\tau_{k-1,1} \quad (6)$$

$$\Delta T_{k,1} = \Delta T_{k-1} + \zeta\Delta\tau_{k-1,1} \quad (7)$$

$$\tau_{k,2} = \tau_{k-1} + \gamma\Delta\tau_{k-1,2} + \Delta T_{k-1} + \zeta\Delta\tau_{k-1,2} \quad (8)$$

$$\Delta T_{k,2} = \Delta T_{k-1} + \zeta\Delta\tau_{k-1,2} \quad (9)$$

When there is one register stage within the loop filter as shown in Fig. 16, the truth table of the select signal is as shown in Table II, where  $\Theta := (\Gamma_{k+1}(0) \leq \Gamma_{k+1}(1))$  and where  $\neg$  is the logical-complement operation.

Therefore, the select signal

$$s_{k+1} = (\neg\hat{\chi}_{k+1}(\arg\min_{\chi_{k+1}} \Gamma_k(\chi_{k+1})) \wedge \hat{\chi}_k(0)) \vee (\hat{\chi}_{k+1}(\arg\min_{\chi_{k+1}} \Gamma_k(\chi_{k+1})) \wedge \hat{\chi}_k(1)). \quad (10)$$

can be calculated as a simple logic function.

When there are two register stages within the loop filter, the

select signal can be calculated as follows.

$$\begin{aligned}
s_{k+2} = & (\hat{\chi}_{k+2}(\operatorname{argmin}_{\chi_{k+2}} \Gamma_{k+2}(\chi_{k+2})) \wedge \hat{\chi}_{k+1}(1) \wedge \hat{\chi}_k(1)) \\
& \vee (\hat{\chi}_{k+2}(\operatorname{argmin}_{\chi_{k+2}} \Gamma_{k+2}(\chi_{k+2})) \wedge \neg \hat{\chi}_{k+1}(1) \wedge \hat{\chi}_k(0)) \\
& \vee (\neg \hat{\chi}_{k+2}(\operatorname{argmin}_{\chi_{k+2}} \Gamma_{k+2}(\chi_{k+2})) \wedge \hat{\chi}_{k+1}(0) \wedge \hat{\chi}_k(1)) \\
& \vee (\neg \hat{\chi}_{k+2}(\operatorname{argmin}_{\chi_{k+2}} \Gamma_{k+2}(\chi_{k+2})) \wedge \neg \hat{\chi}_{k+1}(0) \wedge \hat{\chi}_k(0))
\end{aligned} \tag{11}$$

With each register stage added to the loop filter, the logic circuitry calculating the select signal  $s_k$  gets more complicated, the reliability of the speculative tentative symbol decisions improve, and the loop latency increases by one clock period. The longest path is thereby shortened, and an increase in the data rate is hence achieved. The symbol decisions  $\hat{u}_{k-1}$  are output from the PMU whereas the symbol decisions  $\hat{u}_{k-2}$  are retrieved from the SMU. There is no loss in the reliability of the symbol decisions because they are being updated in the SMU while the loop filter operates. The above description is for the minimal embodiment. The signal  $c_{k-2}$  determining which candidate path metric becomes the path metric would also be communicated to the multiplexers following the register placed after the loop filter in an embodiment wherein the symbol decisions corresponding to the candidate path metrics are output from the PMU in order to decrease the latency in the timing-recovery loop. The register-exchange method as a memory-organization method is ideal for timing recovery because of its low latency.

#### IV. HIGH-SPEED 4-D 5-PAM TCM-DECODER DESIGN

Concatenated RS coding and 4-D 5-PAM TCM has the potential for being used in a future Ethernet standard [30]. In this section, we propose novel methods to calculate branch metrics and path metrics in a high-speed 4-D 5-PAM TCM decoder with eight states [31]. For simplicity, we assume that the ISI is entirely mitigated by means other than embedded per-survivor decision feedback such as Tomlinson–Harashima precoding so that the branch-metric and path-metric calculations are state-independent. A modified Viterbi algorithm [32], [33] similar to the one used by TCM was proposed to deal with 2-D ISI.

##### A. Branch-Metric Calculations

Calculating 1-D branch metrics  $\{\lambda_k(A_0, j), \lambda_k(A_1, j)\}$  requires resolving parallel branch metrics within the 1-D subsets  $\{A_0, A_1\}$ , respectively, for dimension  $j$ , where  $j \in \{0, 1, 2, 3\}$ . The function  $f_d(z_k(j), A(i))$ , where  $i \in \{0, 1, 2, 3, 4\}$ , calculates the distance of  $z_k(j)$  from  $A(i)$  by a distance measure such as the SED and ED. For convenience, we adopt the shorthand notation  $f_d(i)$  to denote  $f_d(z_k(j), A(i))$ . The 1-D branch-metric calculation for the 1-D subset  $A_0$

$$\lambda_k(A_0, j) = \min(f_d(0), f_d(2), f_d(4)) \tag{12}$$

can be mapped into a circuit diagram shown in Fig. 17a, whose propagation delay is the sum of that of one distance calculator performing  $f_d$ , one digital comparator, and one 3-to-1 multiplexer. For a distance measure, such as the SED and ED, the following holds:

$$\lambda_k(A_0, j) = \begin{cases} f_d(0), & \text{if } z_k(j) \leq A(1) \\ f_d(2), & \text{if } A(1) < z_k(j) \leq A(3) \\ f_d(4), & \text{otherwise,} \end{cases}$$

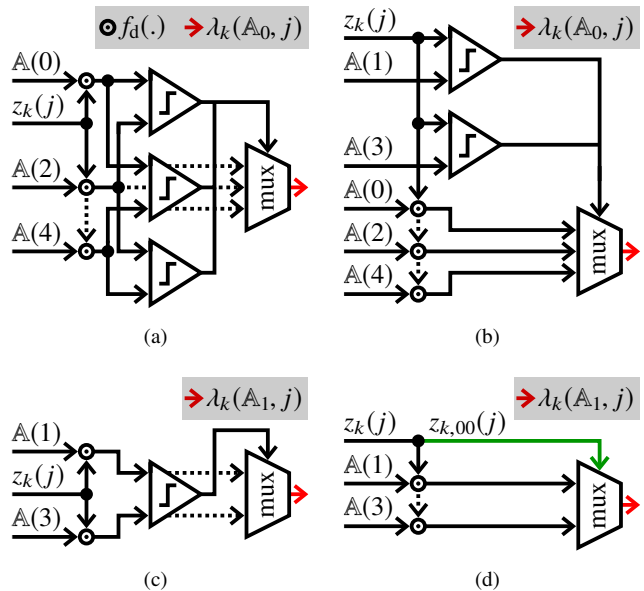


Fig. 17. Conventional (a) and proposed (b) circuit diagrams for calculating the 1-D branch metric  $\lambda_k(A_0, j)$  for the 1-D subset  $A_0$ . Conventional (c) and proposed (d) circuit diagrams for calculating the 1-D branch metric  $\lambda_k(A_1, j)$  for the 1-D subset  $A_1$ .

resulting in a circuit diagram shown in Fig. 17b, whose propagation delay is the sum of that of one distance calculator performing  $f_d$  and one 3-to-1 multiplexer. The reason the propagation delay of the digital comparator no longer contributes to that of the circuit diagram is because it is typically shorter than that of a distance calculator performing  $f_d$ ; for instance, in an advanced technology node such as the 14-nm CMOS. It can also be observed from Fig. 17a and Fig. 17b that the latter needs only two digital comparators whereas the former needs three. We therefore can formulate the 1-D branch-metric  $\lambda_k(A_0, j)$  calculation for the 1-D subset  $A_0$  in such a way as to reduce the area, power consumption, and propagation delay of its design.

Similarly, the 1-D branch-metric calculation for the 1-D subset  $A_1$

$$\lambda_k(A_1, j) = \min(f_d(1), f_d(3)) \tag{13}$$

can be mapped into a circuit diagram shown in Fig. 17c, whose propagation delay is the sum of that of one distance calculator performing  $f_d$ , one digital comparator, and one 2-to-1 multiplexer. For a distance measure, such as the SED and ED, the following holds:

$$\lambda_k(A_1, j) = \begin{cases} f_d(1), & \text{if } z_k(j) \leq A(2) \\ f_d(3), & \text{otherwise,} \end{cases}$$

resulting in a circuit diagram shown in Fig. 17d, whose propagation delay is the sum of that of one distance calculator performing  $f_d$  and one 2-to-1 multiplexer. It can also be observed from Fig. 17c and Fig. 17d that the latter does not need a digital comparator whereas the former does. We therefore can formulate the 1-D branch-metric  $\lambda_k(A_1, j)$  calculation for the 1-D subset  $A_1$  in such a way as to reduce the area, power consumption, and propagation delay of its design.

The same reformulation can be adopted for calculating 4-D branch metrics, which results in a significantly reduced area,

**Algorithm 3:** Path-metric update for the trellis in Fig. 5b.

---

**Input:**  $\Gamma_k(x_k, x_{k+1})$ ;  
**Output:**  $\Gamma_{k+1}(x_{k+1})$ ;

*Initialization*

1:  $\tau \leftarrow 0$ ;

*Candidate path-metric update*

2: **for**  $k = 0$  to  $K - 1$  **do**

3:   **for**  $x_{k+1} = 0$  to  $7$  **do**

4:      $\tau \leftarrow 0$ ;

5:     **for**  $x_k \in \{\lfloor x_{k+1}/4 \rfloor, \lfloor x_{k+1}/4 \rfloor + 4\}$  **do**

6:        $\zeta_k \leftarrow \Gamma_k(x_k, x_{k+1})$ ;

7:        $\phi_k \leftarrow \Gamma_k(x_k + 2, x_{k+1})$ ;

8:        $\Gamma_{k+1}^\tau(x_{k+1}) \leftarrow \text{pathMetricComparison}(\zeta_k, \phi_k)$ ;

9:        $\tau \leftarrow \tau + 1$ ;

10:     **end for**

11:   **end for**

*Path-metric update after pipelining*

12: **for**  $x_k = 0$  to  $7$  **do**

13:    $\zeta_k \leftarrow \Gamma_k^0(x_k)$ ;

14:    $\phi_k \leftarrow \Gamma_k^1(x_k)$ ;

15:    $\Gamma_k(x_k) \leftarrow \text{pathMetricComparison}(\zeta_k, \phi_k)$ ;

16: **end for**

17: **end for**

*Comparison of two partial or candidate path metrics*

18: **function**  $\text{pathMetricComparison}(\zeta_k, \phi_k)$

19: **if**  $(\zeta_{k,1N} \leq \phi_{k,1N})$  **then**

20:    $c_{k,\text{amp}}(\zeta_k, \phi_k) \leftarrow 1$ ;

21: **else**

22:    $c_{k,\text{amp}}(\zeta_k, \phi_k) \leftarrow 0$ ;

23: **end if**

24: **if**  $(\zeta_{k,00} = \phi_{k,00})$  **then**

25:    $c_{k,\text{sel}}(\zeta_k, \phi_k) \leftarrow c_{k,\text{amp}}(\zeta_k, \phi_k)$ ;

26: **else**

27:    $c_{k,\text{sel}}(\zeta_k, \phi_k) \leftarrow \neg c_{k,\text{amp}}(\zeta_k, \phi_k)$ ;

28: **end if**

29: **if**  $(c_{k,\text{sel}}(\zeta_k, \phi_k) = 1)$  **then**

30:    $\Gamma_{k+1} \leftarrow \zeta_k$ ;

31: **else**

32:    $\Gamma_{k+1} \leftarrow \phi_k$ ;

33: **end if**

34: **return**  $\Gamma_{k+1}$ ;

---

power consumption, propagation delay of the longest path of the BMU of a TCM decoder.

### B. Path-Metric Calculations

For simplicity, we henceforth omit from our notation the dimension  $j$ .

1) *Algorithm:* Let us suppose that it is sufficient to represent the partial path metrics  $\Gamma(x_k, x_{k+1})$  in  $N$  bits [34], [35]. We extend the partial path metrics by one bit; i.e.,  $\Gamma(x_k, x_{k+1}) = (a_0, a_1, \dots, a_N)$ , where  $a_0$  is the extra bit,  $a_1$  is the most significant bit, and  $a_N$  is the least significant bit. All path metrics are set to the same value, typically 0, at time  $k = 0$ ; i.e.,  $\Gamma(x_k) = (0, 0, \dots, 0)$ . In this case, regardless of whether the 2's-complement or unsigned binary representation is used, unlike in [34] where the 2's-complement representation must be used, the extra bit  $a_0$

is set to 0 at time  $k = 0$ . For the trellis diagram given in Fig. 5b, there are four competing partial path metrics to become the path metric in the next time step for a given state; e.g.,  $\Gamma_{k+1}(0) = \min(\Gamma_k(0, 0), \Gamma_k(2, 0), \Gamma_k(4, 0), \Gamma_k(6, 0))$ . Let us define

$$\Gamma_{k,0i}(x_k, x_{k+1}) = (g_0, \dots, g_i) \quad \forall i \in \mathbb{Z} : 0 \leq i \leq N. \quad (14)$$

Then, for a state  $x_{k+1}$  in the trellis in Fig. 5b, the sign-comparison function is

$$c_{k,\text{sgn}}(\zeta_k, \phi_k) = \begin{cases} 0, & \text{if } \zeta_{k,00} = \phi_{k,00} \\ 1, & \text{otherwise,} \end{cases}$$

where both  $\zeta_k$  and  $\phi_k$  represent a partial path metric  $\Gamma_k(x_k, x_{k+1})$  corresponding to an allowed transition from the state  $x_k$  to the state  $x_{k+1}$  or a candidate path metric  $\{\Gamma_{k+1}^0(x_k), \Gamma_{k+1}^1(x_k)\}$ . Next, let us define the amplitude-comparison function as follows:

$$c_{k,\text{amp}}(\zeta_k, \phi_k) = \begin{cases} 1, & \text{if } \zeta_{k,1N} \leq \phi_{k,1N} \\ 0, & \text{otherwise.} \end{cases}$$

The wrap-around select function is then the following:

$$c_{k,\text{sel}}(\zeta_k, \phi_k) = \begin{cases} c_{k,\text{amp}}(\zeta_k, \phi_k), & \text{if } c_{k,\text{sgn}}(\zeta_k, \phi_k) = 0 \\ \neg c_{k,\text{amp}}(\zeta_k, \phi_k), & \text{otherwise.} \end{cases}$$

A candidate path metric is chosen as follows:

$$\Gamma_{k+1}^0(x_{k+1}) = \begin{cases} \zeta_k, & \text{if } c_{k,\text{sel}}(\zeta_k, \phi_k) = 1 \\ \phi_k, & \text{otherwise.} \end{cases}$$

The other candidate path metric  $\Gamma_{k+1}^1(x_{k+1})$  is computed in parallel by using the corresponding partial path metrics  $\Gamma_k(x_k, x_{k+1})$ . The path metric  $\Gamma_{k+1}(x_{k+1})$  is finally chosen by applying the candidate path metric  $\Gamma_{k+1}^0(x_{k+1})$  and  $\Gamma_{k+1}^1(x_{k+1})$  as inputs  $\zeta_k$  and  $\phi_k$  to the procedure described above. The complete algorithm is given in Algorithm 3, which also accounts for a pipeline stage to decrease the propagation delay of the longest path of the PMU. We note that line 5 of Algorithm 3 is for calculations corresponding to state transitions allowed in the trellis diagram shown in Fig. 5b.

2) *Architecture:* The block diagram of the path-metric calculation given in Algorithm 3 is shown in Fig. 18. The candidate path-metric  $\Gamma_{k-1}^0(x_{k-1})$  calculation is performed by comparing two partial path metrics  $\Gamma_{k-1}(x_{k-1}, x_k)$ , which is highlighted in yellow. Similarly, the candidate path-metric  $\Gamma_{k-1}^1(x_{k-1})$  calculation is performed by comparing the other two partial path metrics  $\Gamma_{k-1}(x_{k-1}, x_k)$ , which is highlighted in green. The path-metric  $\Gamma_{k-1}(x_{k-1})$  calculation is performed by comparing two candidate path metrics  $\{\Gamma_{k-1}^0(x_{k-1}), \Gamma_{k-1}^1(x_{k-1})\}$ , which is highlighted in blue.

A pipeline stage follows the yellow- and green-highlighted circuits in order to shorten the propagation delay of the longest path of the PMU, which renders it possible to speculate about the path metrics  $\Gamma_{k-1}(x_{k-1})$  by using the corresponding candidate path metrics  $\{\Gamma_{k-1}^0(x_{k-1}), \Gamma_{k-1}^1(x_{k-1})\}$ . In doing so, the candidate path metrics  $\{\Gamma_{k-1}^0(x_{k-1}), \Gamma_{k-1}^1(x_{k-1})\}$  become immediately available in a clock period at the output of the pipeline stage to be input to the adders at the input of the PMU. The path metric  $\Gamma_{k-1}(x_{k-1})$  is calculated while the addition operation at the input of the PMU is performed. The propagation delay of the longest path is then reduced

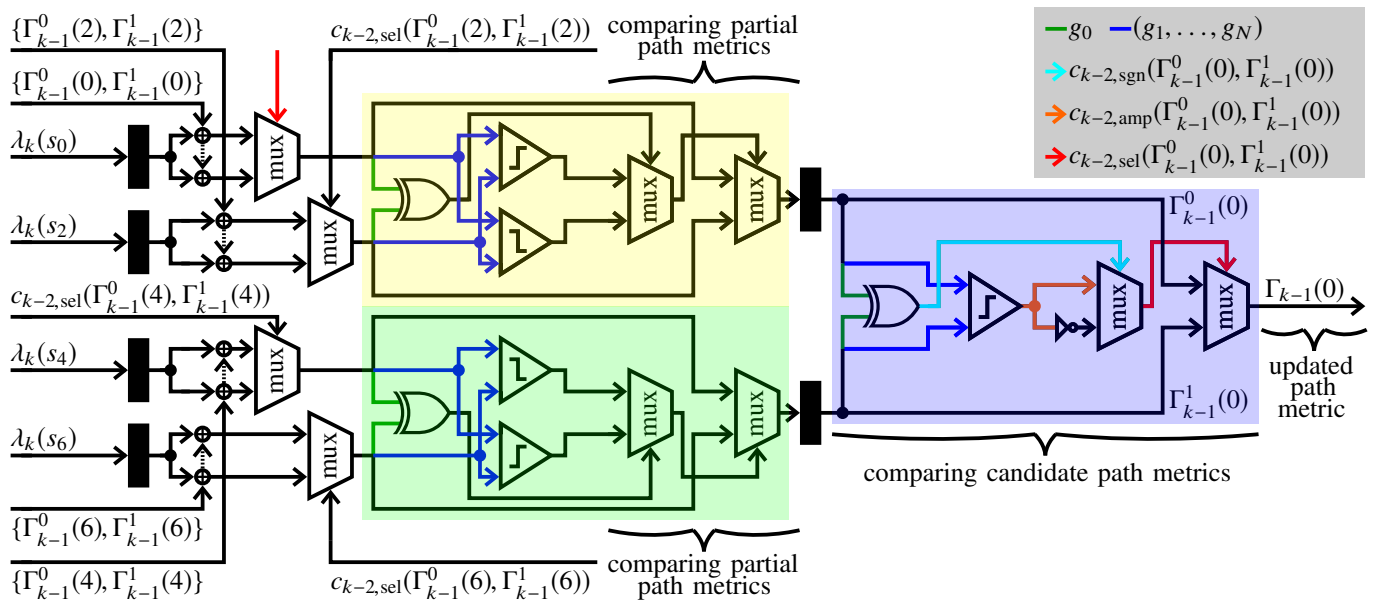


Fig. 18. Block diagram of the proposed sub-PMU for calculating the path metric  $\Gamma_{k-1}(0)$  in the 4-D 5-PAM TCM scheme.

by that of one digital comparator, one inverter, and two 2-to-1 multiplexers because their overall propagation delay is typically smaller than that of an adder; for instance, in an advanced technology node such as the 14-nm CMOS. The inversion of  $c_{k-1,amp}(\Gamma_{k-1}^0(x_{k-1}))$  and  $c_{k-1,amp}(\Gamma_{k-1}^1(x_{k-1}))$  is performed in the yellow- and green-highlighted circuits by inverting the inputs of the respective digital comparators in order to avoid the contribution of the propagation delay of one inverter to that of the longest path in the PMU.

## V. CONCLUSION

This paper presented a 4-PAM reduced-state sliding-block VD prototype with two substates and two embedded per-survivor decision-feedback taps. It was measured to be operating at a data rate of 25.6 Gb/s with an energy efficiency of 4.1 pJ/b at a supply of 0.7 V. A data rate of 30.4 Gb/s is also achieved with an energy efficiency of 5.3 pJ/b at a supply of 0.8 V. The chip area is  $1 \times 1 \text{ mm}^2$ . The feasibility of a high-speed sequence detector for a transmission system employing a bandwidth-efficient modulation scheme was thereby demonstrated, in terms of achievable active-area, energy-efficiency, latency, and speed figures, which are comparable to those of symbol-by-symbol decision solutions. This work is the highest speed multi-level VD with the best energy efficiency and smallest active area reported to date. This result was achieved by a new pipelined RSSD algorithm where the computation of partial path metrics is performed with only one addition in the longest path. A novel speculative symbol timing recovery scheme was also proposed for high-speed data transmission to reduce the latency in the timing-recovery loop.

To date, all the 50-Gb/s to 100-Gb/s per lane high-speed interconnects including Ethernet, Fibre Channel, and Infini-Band have selected 4-PAM as their modulation scheme. The proposed RSSD design is therefore well suited for 4-PAM applications. A new concatenated RS coding and 4-D 5-PAM TCM scheme, which has the potential for being used in future

high-speed interconnect standards, was also introduced. The proposed high-speed 4-D 5-PAM TCM decoder design is suitable for the envisaged future applications.

## REFERENCES

- [1] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
- [2] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [3] H. Kröll *et al.*, "An evolved GSM/EDGE baseband ASIC supporting Rx diversity," *IEEE J. Solid-State Circuits*, vol. 50, no. 7, pp. 1690–1701, Jul. 2015.
- [4] S. Song *et al.*, "A maximum-likelihood sequence detection powered ADC-based serial link," *IEEE Trans. Circuits Syst.*, vol. PP, no. 99, pp. 1–10, Dec. 2017.
- [5] H. Yueksel *et al.*, "A 3.6pJ/b 56Gb/s 4-PAM receiver with 6-bit TI-SAR ADC and quarter-rate speculative 2-tap DFE in 32 nm CMOS," in *Eur. Solid-State Circuits Conf.*, 2015, pp. 148–151.
- [6] M. V. Eyuboglu and S. U. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 13–20, Jan. 1988.
- [7] P. R. Chevillat and E. Eleftheriou, "Decoding of trellis-encoded signals in the presence of intersymbol interference and noise," *IEEE Trans. Commun.*, vol. 37, no. 7, pp. 669–676, Jul. 1989.
- [8] M. V. Eyuboglu and S. U. Qureshi, "Reduced-state sequence estimation for coded modulation on intersymbol interference channels," *IEEE J. Selected Areas Commun.*, vol. 7, no. 6, pp. 989–995, Aug. 1989.
- [9] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.
- [10] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 428–436, May 1989.
- [11] P. J. Black and T. H. -Y. Meng, "A 1-Gb/s, four-state, sliding block Viterbi decoder," *IEEE J. Solid-State Circuits*, vol. 32, no. 6, pp. 797–805, Jun. 1997.
- [12] G. Fettweis and H. Meyr, "High-rate Viterbi processor: a systolic array solution," *IEEE J. Selected Areas Commun.*, vol. 7, no. 8, pp. 1520–1534, Oct. 1990.
- [13] *IEEE Standard for Ethernet Amendment: Physical Layer Specifications and Management Parameters for 100 Gb/s Operation Over Backplanes and Copper Cables*, IEEE Standard 802.3bj, 2015.
- [14] R. D. Cideciyan, M. Gustlin, M. P. Li, J. Wang, and Z. Wang, "Next generation backplane and copper cable challenges," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 130–136, Dec. 2013.

- [15] *Physical Coding Sublayer, Physical Medium Attachment Sublayer and Baseband Medium, Type 1000BASE-T*, IEEE Standard 802.3ab, 2015.
- [16] 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 *Eur. Solid-State Circuits Conf.*, 2016, pp. 309–312.
- [17] 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 *Int. Conf. Signal Process. Commun. Syst.*, 2016, pp. 1–6.
- [18] M. Kossel *et al.*, "A 10 Gb/s 8-tap 6b 2-PAM/4-PAM Tomlinson-Harashima precoding transmitter for future memory-link applications in 22-nm SOI CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3268–3284, Dec. 2013.
- [19] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [20] IEEE P802.3bs 400 GbE Task Force, "Channel Data," 2014. [Online]. Available: <http://www.ieee802.org/3/bs/public/channel/index.shtml>
- [21] B. Zand and D. A. Johns, "High-speed CMOS analog Viterbi detector for 4-PAM partial-response signaling," *IEEE J. Solid-State Circuits*, vol. 37, no. 7, pp. 895–903, Jul. 2002.
- [22] N. Bruels *et al.*, "A 2.8 Gb/s, 32-state, radix-4 Viterbi decoder add-compare-select unit," in *IEEE Symp. VLSI Circuits*, pp. 170–173, 2004.
- [23] 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.
- [24] 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.
- [25] T. Veigel, T. Alpert, F. Lang, M. Grözing, and M. Berroth, "A Viterbi equalizer chip for 40 Gb/s optical communication links," in *Eur. Microwave Integrated Circuits Conf.*, pp. 49–52, 2013.
- [26] *Process Integration, Devices, and Structures*, 2003 ed., International Technology Roadmap for Semiconductors, 2003, pp. 1–37.
- [27] *Executive Report*, 2003 ed., International Technology Roadmap for Semiconductors, 2003, pp. 1–65.
- [28] *Executive Report*, 2015 ed., International Technology Roadmap for Semiconductors 2.0, 2015, pp. 1–69.
- [29] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt and W. Schott, "A PRML system for digital magnetic recording," *IEEE J. Selected Areas Commun.*, vol. 10, no. 1, pp. 38–56, Jan. 1992.
- [30] H. Yueksel *et al.*, "High-speed link with trellis-coded modulation and Reed-Solomon coding," in *IEEE Conf. Standards for Commun. Networking*, 2016, pp. 1–6.
- [31] H. Yueksel, "High-speed wireline link design," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2017.
- [32] S. Nabavi, B. V. K. Vijaya Kumar, and J.-G. Zhu, "Modifying Viterbi algorithm to mitigate intertrack interference in bit-patterned media," *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 2274–2276, Jun. 2007.
- [33] Y. Wang and B. V. K. Vijaya Kumar, "Bidirectional decision feedback modified Viterbi detection (BD-DFMV) for shingled bit-patterned magnetic recording (BPMR) with 2D sectors and alternating track widths," *IEEE J. Selected Areas Commun.*, vol. 34, no. 9, pp. 2450–2462, Sep. 2016.
- [34] C. B. Shung, P. H. Siegel, G. Ungerboeck, and H. K. Thapar, "VLSI architectures for metric normalization in the Viterbi algorithm," in *IEEE Int. Conf. Commun.*, pp. 1723–1728, 1990.
- [35] A. P. Hekstra, "An alternative to metric rescaling in Viterbi decoders," *IEEE Trans. Commun.*, vol. 37, no. 11, pp. 1220–1222, Nov. 1989.



**Hazar Yueksel** (S'08-M'17) received the degrees of B.Sc. in electrical and electronics engineering from Bilkent Üniversitesi, Turkey, in 2010, M.Sc. *cum Laude* in electronics engineering from the Politecnico di Milano, Italy, in 2012, and Ph.D. in electrical engineering from the École Polytechnique Fédérale de Lausanne, Switzerland, in 2017.

He joined IBM Research – Zurich, Switzerland, in 2013, where he conducted research for his doctoral dissertation into algorithms and architectures for high-speed, high-performance, low-power, low-

complexity, low-latency transceiver implementations. He worked as a postdoctoral research scientist at Columbia University, NY, USA after his doctoral studies. He is currently a research scientist at the IBM T. J. Watson Research Center, NY, USA.



**Matthias Braendli** received the Dipl. Ing. (M.Sc.) degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, in 1997.

From 1998 to 2001, he was with the Integrated Systems Laboratory of the Swiss Federal Institute of Technology, working on deep-submicron technology VLSI design challenges, digital video image processing for biomedical applications, and testability of CMOS circuits. In 2001 he joined the Microelectronics Design Center of ETH Zurich, where he was

involved in numerous digital and mixed-signal ASIC design projects, worked on EDA design automation, and contributed to teaching. In 2008, he joined IBM Research – Zurich, Rüschlikon, Switzerland, where he works on multi-gigabit/s, low-power communication circuits in advanced CMOS technologies.



**Andreas Burg** (S'97-M'05) was born in Munich, Germany, in 1975. He received the Dipl.-Ing. degree from ETH Zurich, Zurich, Switzerland, in 2000, and the Dr.Sc.Techn. degree from the Integrated Systems Laboratory of ETH Zurich, in 2006.

In 1998, he was with Siemens Semiconductors, San Jose, CA, USA. During his doctoral studies, he worked at Bell Labs Wireless Research for a total of one year. From 2006 to 2007, he held positions as a Postdoctoral Researcher with the Integrated Systems Laboratory and at the Communication Theory Group

of ETH Zurich. In 2007, he cofounded Celestrius, an ETH-spinoff in the field of MIMO wireless communication, where he was responsible for the ASIC development as Director for VLSI. In January 2009, he joined ETH Zurich as an SNF Assistant Professor and as head of the Signal Processing Circuits and Systems group with the Integrated Systems Laboratory. Since January 2011, he has been a Tenure Track Assistant Professor at the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, where he is leading the Telecommunications Circuits Laboratory.

Mr. Burg was the recipient of the Willi Studer Award and the ETH Medal for his diploma and his diploma thesis, respectively, in 2000. He was also awarded an ETH Medal for his Ph.D. dissertation in 2006. In 2008, he received a four-year grant from the Swiss National Science Foundation (SNF) for an SNF Assistant Professorship. He has served on the TPC of various conferences on VLSI, signal processing, and communications. He was a TPC co-chair for VLSI-SoC 2012 and is a TCP co-chair for SiPS 2017. He served as an editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in 2013 and is on the editorial board of *Microelectronics Journal* and the *MDPI Journal on Low Power Electronics and its Applications*.



**Giovanni Cherubini** (S'80-M'82-SM'94-F'06) received a Laurea degree (*summa cum laude*) from the University of Padova, Italy, in 1981, and M.S. and Ph.D. degrees from the University of California, San Diego, in 1984 and 1986, respectively, all in Electrical Engineering. Since 1987 he has been with the IBM Research Laboratory in Zurich, Switzerland. His research interests comprise high-speed data transmission, data storage, and control systems. He was co-editor of the IEEE 100BASE-T2 Standard for Fast Ethernet transmission over voice-grade cables. More recently, he contributed to the realization of the first atomic-force-microscope-based data-storage prototype. He is currently focusing on advanced servo-control technologies for tape drives and on storage techniques targeting big data applications.

Dr. Cherubini holds over 100 patents in the areas of communication systems, data storage, and control systems, was named Fellow of the IEEE in 2006, and Master Inventor at IBM in 2009 and 2015. He served as Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS from 1999 to 2004. He is currently Senior Editor of the IEEE CONTROL SYSTEMS LETTERS, Associate Editor of the IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS. He was co-recipient of the 2003 Leonard G. Abraham Prize Paper Award of the IEEE Communications Society, the 2009 IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY Outstanding Paper Award, the 2009 Control Systems Technology Award of the IEEE Control Systems Society, the 2011 IBM Research Pat Goldberg Memorial Best Paper Award, and the 2014 Industrial Achievement Award of the International Federation of Automatic Control. In 2013 he received an IBM Corporate Award for Tape Technology Leadership for his contributions to Linear Tape Open and Enterprise tape drives.

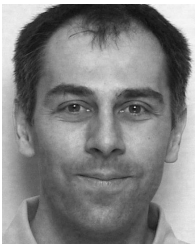




**Roy D. Cideciyan** (S'82-M'86-SM'98-F'10) received the Dipl.-Ing. degree from Aachen University of Technology, Germany, in 1981 and the M.S.E.E. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1982 and 1985, respectively.

In 1986, he joined IBM Research – Zurich, Switzerland, where he has been working in the areas of data storage and data transmission. His research interests include signal processing, coding, detection, synchronization, and performance evaluation

for hard disk, tape, nonvolatile memory, and high-speed links. He holds over 110 US patents and was named Fellow of the IEEE in 2010 for contributions to signal processing and constrained coding for magnetic recording. In 2011, he was co-recipient of the IBM Research Pat Goldberg Memorial Best Paper Award. He designed the transcoding scheme to a novel 256B/257B line code and contributed to the design and analysis of the Reed–Solomon forward error correction (FEC) coding schemes in the IEEE 802.3bj-2014 standard for 100 Gb/s Ethernet over backplane and twinax cables. The line and FEC coding schemes in the IEEE 802.3bj-2014 standard were adopted by high-speed interconnect standards that specify transmission at 25 to 100 Gb/s per lane, including fiber channel standards 32GFC, 64GFC, 128GFC, and 256GFC; InfiniBand EDR and HDR; IEEE 802.3bm for 100 Gb/s Ethernet over multi-mode fibers; IEEE 802.3by for 25 Gb/s Ethernet over backplane, twinax cable and multi-mode fiber; and IEEE 802.3bs for 400 Gb/s Ethernet over multi-mode and single-mode fibers.



**Pier Andrea Francese** (M'01) received a degree in electrical engineering (cum laude) from the Politecnico di Milano, Italy, and the Ph.D. degree from the Federal Institute of Technology of Zurich (ETH), Switzerland, in 1993 and 2005, respectively.

He worked in the field of IC product development for Teradyne, Philips Semiconductors and National Semiconductor. In 2010 he joined IBM Research – Zurich, Rüschlikon, Switzerland, where he is currently designing circuits for high-speed I/O links in advanced CMOS technologies.



**Simeon Furrer** (M'00-SM'15) received the Dipl.-Ing. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology, Zurich, Switzerland, in 1999 and 2005, respectively.

He was a Pre-Doctoral Researcher of Short-Range Radio Communication and Sensor Networks with IBM Research - Zurich, in Rüschlikon, Switzerland. He was a Principal Scientist with Broadcom Corporation, Sunnyvale, CA, USA, from 2005 to 2010, where he was involved in the design and architecture of IEEE 802.11n-compliant multiple input multiple

output wireless local-area network chipsets. Dr. Furrer joined the Cloud and Computing Infrastructure Department of IBM Research Zurich in 2010 as a Research Staff Member. He is currently focusing on advanced signal-processing algorithms for tape read channels and servo control aspects to improve storage capacity. His research interests include data communication and storage systems, signal processing, detection, and estimation.



**Marcel Kossel** (S'99-M'02-SM'09) received the Dipl. Ing. and Ph.D. degrees in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, in 1997 and 2000, respectively.

He joined IBM Research – Zurich in 2001, where he is involved in analog circuit design for high-speed serial links. His research interests include analog circuit design and RF measurement techniques. He has also worked in the field of microwave tagging systems and radio-frequency identification systems.



**Lukas Kull** (S'10-M'14) received the M.Sc. degree in electrical engineering from the Swiss Federal Institute of Technology, Zurich (ETH), Switzerland, in 2007 and the Ph.D. degree from the Swiss Federal Institute of Technology, Lausanne (EPFL), Switzerland, in 2014.

He joined IBM Research – Zurich, Rüschlikon, Switzerland, in 2010, where he has been involved in analog circuit design for high-speed low-power ADCs. His research interests include analog circuit design, IR and THz imaging. In these areas he

authored or co-authored more than 10 patents and 20 technical publications.



**Danny Luu** (S'17) received the B.Sc. and M.Sc. degrees in electrical engineering and information technology from the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, in 2013. He is currently pursuing the doctoral degree at ETH Zurich.

He joined IBM Research – Zurich, Rüschlikon, Switzerland, in 2013, where he has been conducting research into analog circuit design for high-speed, high-resolution, low-power ADCs in collaboration with ETH Zurich towards his doctoral degree.



**Christian Menolfi** (S'97-M'99) received the Dipl. Ing. degree and the Ph.D. degree in electrical engineering from the Swiss Federal Institute of Technology (ETH), Zurich, in 1993 and 2000, respectively.

From 1993 to 2000, he was with the Integrated Systems Laboratory, ETH Zurich, as a Research Assistant, where he worked on highly sensitive CMOS VLSI data-acquisition circuits for silicon-based microsensors. Since 2000, he has been with IBM Research – Zurich, Rüschlikon, Switzerland, where he is involved in the design of multi-gigabit

low-power communication circuits in advanced CMOS technologies.



**Thomas Morf** (S'89-M'96-SM'09) received his B.Sc. from the Zurich University of Applied Science, Switzerland, his M.Sc. from the University of California at Santa Barbara (UCSB) and his Ph.D. from the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland.

From 1996 to 1999, he led a research group in the area of InP-HBT circuit design and technology at the ETH. In 1999, he joined IBM Research – Zurich, Rüschlikon, Switzerland. His current research interests include ESD circuit protection, electrical and

optical high-speed high-density interconnects, and THz antennas and detectors. Dr. Morf is senior member of the IEEE and has co-authored more than 80 papers.



**Thomas Toifl** (S'97-M'99-SM'09) received the Dipl.-Ing. (M.Sc.) degree and the Ph.D. degree (with highest honors) from Vienna University of Technology, Austria, in 1995 and 1999, respectively.

In 1996, he joined the Microelectronics Group of the European Research Center for Particle Physics (CERN), Geneva, Switzerland, where he developed radiation-hard circuits for detector synchronization and data transmission, which were integrated in the four particle detector systems of the new Large Hadron Collider (LHC).

In 2001, he joined IBM Research – Zurich in Rüschlikon, Switzerland, where he has been working on multi-gigabit per second, low-power communication circuits in advanced CMOS technologies. In that area, he authored or co-authored 14 patents and numerous technical publications. Since July 2008, he manages the I/O Link Technology group at IBM Research – Zurich.

Dr. Toifl received the Beatrice Winner Award for Editorial Excellence at the 2005 IEEE International Solid-State Circuits Conference (ISSCC).