

# Design and Implementation of an On-Demand Maximum-Likelihood Sequence Estimation (MLSE)

MOHAMMAD EMAMI MEYBODI<sup>1</sup> (Graduate Student Member, IEEE), HECTOR GOMEZ<sup>1</sup> (Member, IEEE),  
YU-CHUN LU<sup>2</sup> (Member, IEEE), HOSSEIN SHAKIBA<sup>3</sup> (Senior Member, IEEE),  
AND ALI SHEIKHOLESLAMI<sup>1</sup> (Senior Member, IEEE)

<sup>1</sup>Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada

<sup>2</sup>Huawei Technologies Company Ltd., Beijing 100095, China

<sup>3</sup>Huawei Technologies Canada, Markham, ON L3R 5A4, Canada

This article was recommended by Guest Editor C. M. Hung.

CORRESPONDING AUTHOR: M. E. MEYBODI (e-mail: m.meybodi@mail.utoronto.ca)

This work was supported in part by the Huawei Technologies Canada and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

**ABSTRACT** This paper proposes a novel design for Maximum Likelihood Sequence Estimation (MLSE) used in ultra-high-speed wireline communication. We take advantage of the propagated errors caused by Decision-Feedback Equalizer (DFE) to activate and guide the MLSE, thereby reducing its complexity. The design is customized for a 4-PAM, 1 + D signaling system, and synthesized in 16nm FinFET TSMC Technology. For comparison purposes, a conventional MLSE is also synthesized in the same technology. The synthesis report confirms that the proposed design consumes 1/10 of the power and occupies 1/15 of the area required by the conventional MLSE while having a comparable bit error rate.

**INDEX TERMS** DFE burst error, MLSE, MLSE on demand, partial equalization, SERDES, wireline.

## I. INTRODUCTION

WITH the steady growth of data rate in wireline communication, the design of transceivers to effectively compensate for high-frequency channel losses has become extremely challenging. The receivers supporting more than 56Gb/s often utilize an analog-to-digital converter (ADC) after the analog front-end (AFE) to provide better equalization in the digital domain [1]–[3]. This makes transmission of data over long-reach channels with more than 40dB loss at the Nyquist frequency feasible at the expense of more power consumption [4]–[9]. The recent state-of-the-art designs confirm that 30 – 50% of the total power is consumed in the digital section [2], [7], [9], [10]. Therefore, any attempt at reducing the complexity of the digital equalizers would have a great impact on the total power consumption. Feed-forward equalizers (FFE) and Decision-feedback equalizers (DFEs) are the most common equalizers for cancelling inter-symbol interference (ISI). Even though FFEs degrade the SNR, a large tap-count FFE followed by a

1- or 2-tap DFE is a common approach for equalization in the receiver [4]–[10]. For instance, LaCroix *et al.* [6] use 25-tap FFE and 2-tap DFE at the receiver to implement a 112Gb/s 4-PAM transceiver, achieving an energy efficiency of 5.9pJ/bit for transmission of data over a channel having 45dB loss. Having limited number of taps for DFE is mainly due to the feedback loop around a regular DFE. Therefore, in order to alleviate the stringent timing constraint, several techniques such as unrolling and unfolding have been utilized in the implementation of a DFE [1], [2], [11]–[15]. These techniques cause the complexity of a DFE to grow exponentially with the number of taps, discouraging the use of large tap-counts DFEs in the receiver. In [2], Kiran *et al.* reduce the complexity of the DFE by partially unrolling it. With this approach, they could implement a 2-tap DFE for 4-PAM modulation consuming half of the power of a conventional DFE at 52Gb/s. A more recent work [16] introduces sliding block DFE (SB-DFE) where the DFE feedback loop is broken, relying on the observation that the DFE output

ultimately converges to the correct output even if the initial symbols are wrong. Therefore, having no feedback loop allows them to implement a 9-tap DFE preceded by a 5-tap FFE for equalizing 4-PAM data at the speed of 112Gb/s. Energy efficiency of 1.5pJ/bit for only the digital part is reported in this architecture.

Despite the innovations listed above, the error propagation of the DFE continues to degrade the performance of the system especially when it is used in conjunction with forward error correction (FEC). The probability of generating error bursts by propagation of errors in the DFE loop is related to the number of taps and their values [17]–[19].

The optimal equalizer which can be used in the receiver side is known as the maximum likelihood sequence estimation (MLSE) [20]. MLSE finds the most likely transmitted sequence rather than making symbol-by-symbol decisions. Since it takes the ISI-induced correlations between the symbols into account, it has superior performance over other types of equalizers such as FFE and DFE. Nevertheless, it has not been widely used in high-speed wireline applications due to its complexity and power consumption, even when its efficient implementation method, the Viterbi algorithm, is employed. One of the common approaches for realizing Viterbi algorithm is known as sliding block architecture [21], [22]. This approach enables the use of pipelining by breaking the feedback loop, but at the cost of requiring pre-training that in turn increases area and power. There also exists a more efficient architecture for Viterbi algorithm which is based on look-ahead technique [23], [24]. Even though the look ahead method leads to a more efficient architecture [24], its area and power consumption may still be deemed unacceptable, especially when the number of states in Viterbi algorithm is increased for more complex signaling. Reduced-state sequence estimation is another technique for lowering the complexity of MLSE by reducing the number of states of the Viterbi algorithm [25], [26].

In this paper, we propose implementing MLSE in conjunction with DFE, where we use the DFE output unless we detect an error, in which case we invoke the MLSE [19]. In this method, the MLSE is used on demand to avoid its power consumption compared to a conventional MLSE. We also use DFE information to reduce the complexity of the MLSE. This results in a significant reduction in both area and power consumption while maintaining the BER performance comparable to the conventional MLSE.

The rest of the paper is organized as follows. In Section II, we briefly analyse the error propagation property of DFE and how we can utilize it to invoke the MLSE. The design and implementation of a conventional MLSE are also explained in this section. The proposed method for reducing the complexity of the MLSE is discussed in Section III. The proposed implementation of the reduced MLSE-on-demand is explained in Section IV. The realized algorithms are compared together in Section V, and finally the paper is concluded.

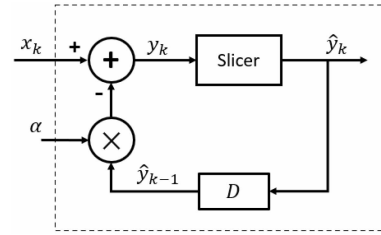


FIGURE 1. 1-tap DFE block diagram.

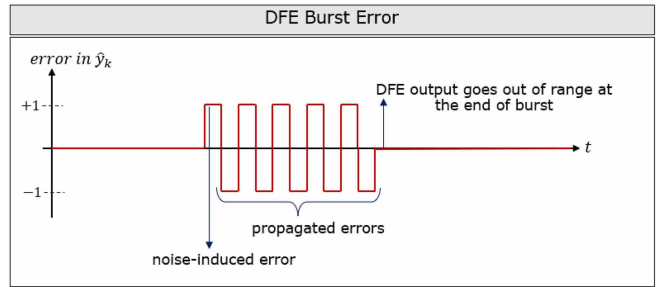


FIGURE 2. DFE error propagation.

## II. BACKGROUND

In this section, we explain the error propagation mechanism of a 1-tap DFE, and investigate methods to cancel these errors. We also review the Viterbi algorithm and its implementation in the hardware for 4-PAM, 1+D signaling.

### A. ERROR PROPAGATION MECHANISM

Fig. 1 shows a block diagram of a 1-tap DFE equalizing  $1 + \alpha D$  channel, where  $x_k$  is the received signal at instant  $k$ .

$$x_k = s_k + \alpha s_{k-1} + n_k \quad (1)$$

where  $s_k$  and  $s_{k-1}$  are the current and previous transmitted symbols, respectively,  $\alpha$  is a positive coefficient and is equal to the strength of the post-cursor and  $n_k$  is the noise amplitude at the current instant.

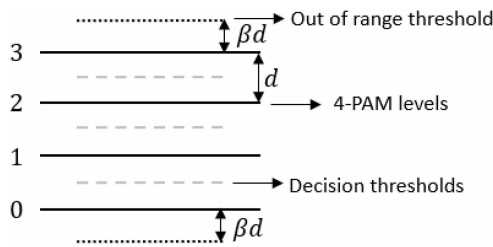
The slicer input,  $y_k$ , is equal to:

$$y_k = s_k + \alpha(s_{k-1} - \hat{y}_{k-1}) + n_k \quad (2)$$

In an ideal world,  $\hat{y}_{k-1} = s_{k-1}$  and  $n_k = 0$ . Hence,  $\hat{y}_k$  would be equal to  $s_k$ . However, due to non-zero noise these two values might be different. Let the difference ( $\hat{y}_k - s_k$ ) be denoted by  $e_k$ . Hence:

$$y_k = s_k - \alpha e_{k-1} + n_k \quad (3)$$

Eq. (3) implies that if there is an error in the detection of the previous symbol, i.e.,  $e_{k-1} = \pm 1$  (The error between the non-adjacent levels is neglected), the current level at the slicer input will move in a direction opposite to that of the previous error. As  $\alpha$  increases, the effect of the previous error is more pronounced at the current slicer input. The tendency toward the opposite direction leads to a zigzag pattern for the error propagation as shown in Fig. 2.



**FIGURE 3.** 4-PAM levels and the thresholds of the slicer's comparators.

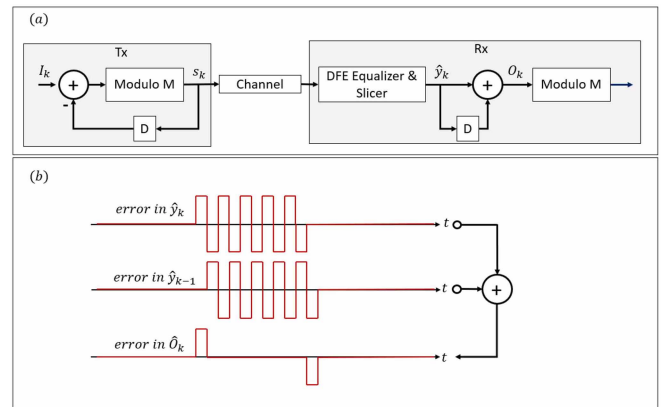
To see how this burst of errors eventually terminates, we refer to Eq. (3) and Fig. 3 drawn for 4-PAM modulation. Based on Eq. (3), the signal level at the slicer input is displaced by  $\alpha$  times the previous error. Since we assumed that the absolute value of the error is 1, the amount of shift is determined by  $\alpha$ . If noise is neglected and  $\alpha < 0.5$ , the displacement is not enough to change the decision of the slicer. Therefore, error is not propagated. On the other hand, if  $\alpha > 0.5$ , the displacement makes the slicer mistake the signal for either one level higher or one level lower than the actual value depending on the polarity of the previous error. However, based on Fig. 3, if the signal ( $s_k$ ) is already at its maximum level (which is 3 for 4-PAM case), one level higher than this value does not change the slicer decision. Similarly, if the signal is at its minimum level, one level lower than this level does not make a difference. Therefore, the burst of errors is terminated in these particular cases where the slicer input goes out-of-range. These out-of-range signals can be an excellent indicator for the end of burst detection. We should note that in real applications, noise slightly changes the probability of error propagation and termination. Nevertheless, both exact analysis and simulation results confirm that its impact is insignificant, especially if  $\alpha$  is not close to 0.5.

### B. PRECODING

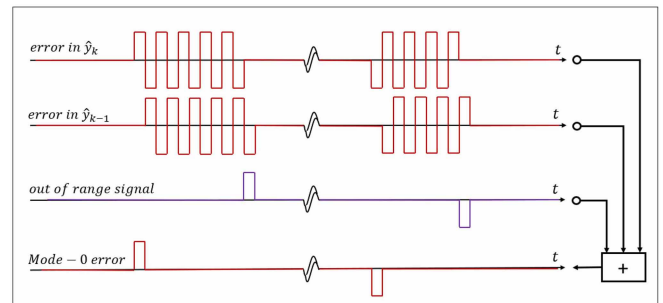
In the previous section, we explained that the burst of the errors in the DFE output alternates between  $-1$  and  $+1$ . Therefore, by adding the current DFE output to the previous one, the net error would become zero, except for the first and the last error in the burst. However, taking the result of the addition as the output requires performing the inverse function at the transmitter (Fig. 4). This technique for removing the intermediate errors in the burst is called precoding [18].

### C. END OF BURST OF ERRORS CORRECTION

As discussed earlier, the burst of errors is terminated when the signal at the slicer input goes out-of-range. The out-of-range signals can be detected by two additional comparators. Hence, by detection of the end of burst, we can also correct the last error in the burst simply by adding or subtracting one level depending on the polarity of the out-of-range signal (Fig. 5). As shown on Fig. 3, the threshold levels of these two additional comparators are set to be  $\beta d$  apart



**FIGURE 4.** (a) Precoding technique for elimination of the burst error (b) Value of the errors at the DFE and the decoder output.

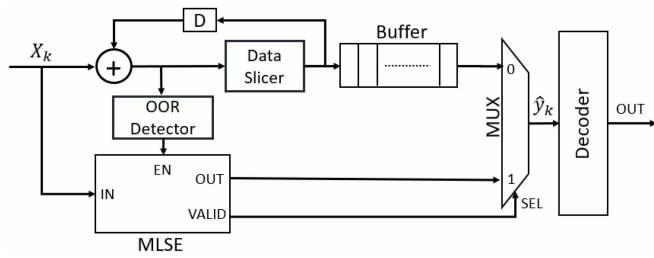
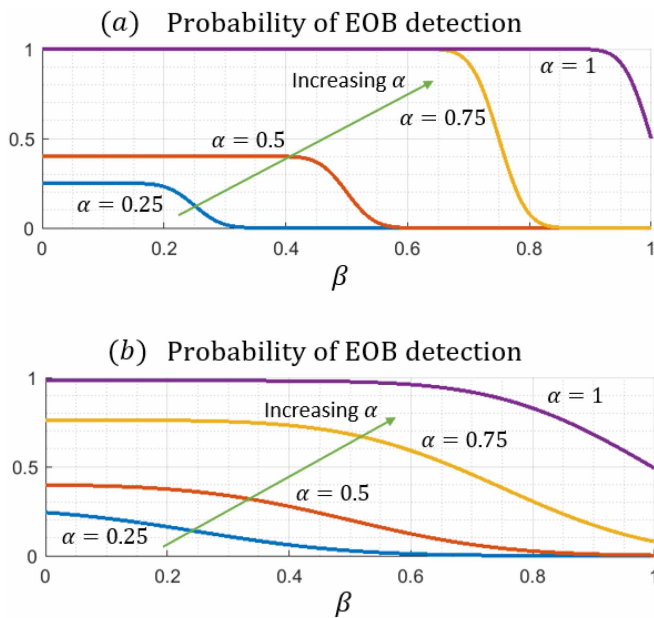


**FIGURE 5.** Mode-0 of the error correction.

from the maximum and minimum levels, where  $d$  is the distance between two adjacent levels and  $\beta$  is a constant to be determined. If  $\beta$  is too low, a small noise can falsely trigger the out-of-range comparators. On the other hand, if it is too high, we may miss the true out-of-range signals. To determine an optimal value, we compute the BER with respect to  $\beta$  and derive  $\beta$  such that the BER is minimized. Following [19], we refer to this method as “mode-0” for error correction. Mode-0, in conjunction with the precoding, is able to remove all the propagated errors in DFE except the first error which occurs due to noise [18].

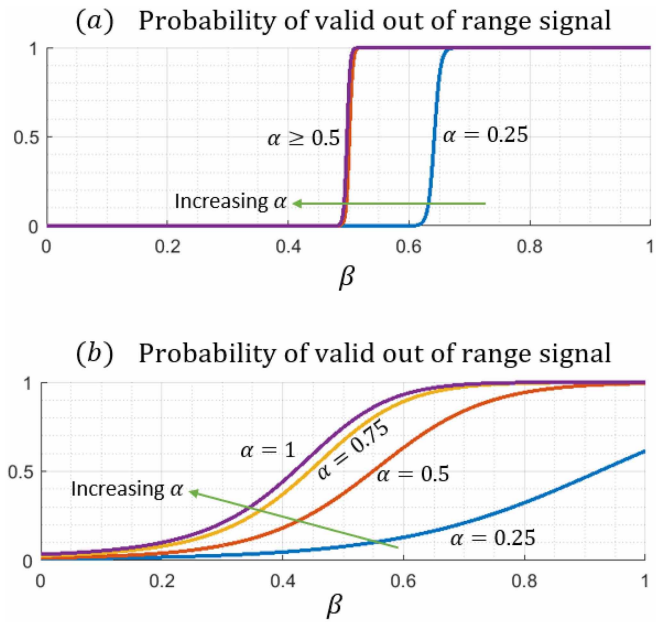
### D. MLSE ON DEMAND

MLSE recovers the most probable sequence of data, and Viterbi algorithm is often used as an optimal algorithm for the MLSE implementation. This algorithm as explained in the next section is too complex and power hungry to be implemented in high data rate applications [20]. However, the idea of end of burst (EOB) of errors detection can be used to activate an MLSE engine for determining the most probable transmitted sequence before the end of burst. This method has been proposed by Lu *et al.* [19], where they demonstrated a significant improvement in BER which even includes correction of the first error of the burst. Fig. 6 depicts a block diagram implementation of the idea. For as long as the out-of-range signal has not been detected, the DFE provides the output. Once an out-of-range signal is


**FIGURE 6.** MLSE on Demand conceptual block diagram.

**FIGURE 7.** Probability of end of burst of errors detection for 4-PAM signaling with (a) High SNR (b) Low SNR.

detected, the MLSE engine is activated to detect the recent transmitted symbols prior to the end of burst. During this time the MLSE provides the output before it switches back to the DFE output after the burst is terminated. To match the delay of the MLSE, a buffer is inserted at the output of the DFE to make the switch seamless. The sequence of symbols on which the MLSE engine works should be long enough to not only include the first error in the burst, but also go enough beyond that to confidently determine the most probable sequence. Finally, the decoder after the multiplexer decodes the symbols that were precoded as discussed previously.

Like mode-0 of the error correction, the performance of this method relies on the value of  $\beta$ . Fig. 7 depicts the probability of detecting EOB with different values of  $\alpha$  and for two different cases of signal-to-noise ratio (SNR). This figure implies that the value of  $\beta$  should be kept below the value of  $\alpha$ , if we do not want to miss a lot of EOBs. Otherwise, the MLSE engine is not activated for the missed EOBs and the BER of the system approaches a DFE-based receiver. However, if  $\alpha \leq 0.5$ , we inevitably miss a lot of EOBs regardless of the value of  $\beta$  and SNR. We should also note


**FIGURE 8.** Probability of the valid out-of-range signals for 4-PAM signaling with (a) High SNR (b) Low SNR.

that the out-of-range detectors might be falsely triggered by noise. Therefore, the out-of-range signals indicate EOB with a likelihood of less than 100%. This probability depends on both  $\alpha$  and  $\beta$ , and it is reflected on Fig. 8 for two different cases of SNR. According to this figure, the value of  $\beta$  should be higher than 0.5 if we desire to avoid frequent false activation of the MLSE engine. In theory, false activation of the MLSE engine would only increase the power consumption and would not degrade BER. However, in practice, we have a limited capacity for handling the out-of-range signals. Therefore, if the system is flooded by the false out-of-range signals, it will not be able to handle the valid ones, resulting in a BER which is close to a DFE-based receiver. In summary, we can say that MLSE-on-demand can be competitive with conventional MLSE, if  $0.5 < \beta < \alpha$ . In this paper, we designed MLSE-on-demand for  $1 + D$  and set the value of  $\beta$  to 0.6.

The key benefit of the MLSE-on-demand method is that complex MLSE is just invoked whenever needed, thereby considerable amount of energy is saved. However, it still requires the same area as the conventional MLSE. In Section III, an algorithm to drastically reduce the area is discussed.

### E. CONVENTIONAL MLSE

In the MLSE, we find the least costly path (a path which incurs the minimum amount of error) through the trellis diagram in order to recover the most probable sequence sent by the transmitter. Fig. 9 demonstrates an example of trellis diagram for 4-PAM signaling assuming one post-cursor ISI with the value of  $\alpha$ . Hence, there exist four states in each step of the trellis diagram, and 16 possible transitions from the current step to the next one.



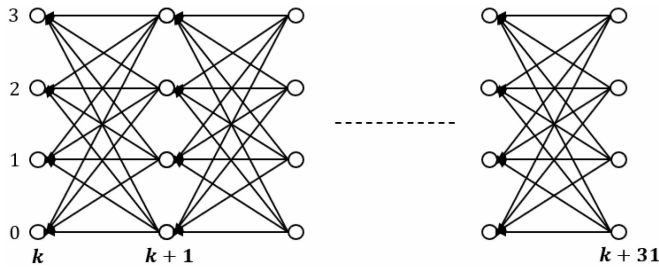


FIGURE 9. Trellis diagram for 4-PAM signaling (window size = 32).

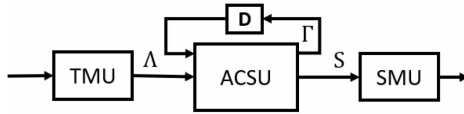


FIGURE 10. Sections of the Viterbi algorithm as implemented in the hardware.

Viterbi algorithm breaks the process of finding the shortest path into three separate sections as shown in Fig. 10 [23]. Those are the transition metric unit (TMU), add compare select unit (ACSU), and survivor memory unit (SMU). TMU computes the cost of transitions existing between two adjacent steps of the trellis diagram. This cost is the power of the error defined by Eq. (4), where  $TM$  is the transition metric,  $y$  is the input signal level, and  $x_{ij}$  is the expected level of the input signal, if the transmitter sends data sequence  $ji$ .

$$TM_{ij} = \lambda_{ij} = (y - x_{ij})^2 \quad (4)$$

ACSU updates the state metrics in the trellis diagram by adding all the transition metrics to the current state metrics and taking the minimum ones as the next values of the state metrics (Eq. (5)).

$$\gamma_{i,k+1} = \min(\gamma_{0,k} + \lambda_{i0,k}, \gamma_{1,k} + \lambda_{i1,k}, \gamma_{2,k} + \lambda_{i2,k}, \gamma_{3,k} + \lambda_{i3,k}) \quad (5)$$

where  $\gamma$  and  $\lambda$  represent state metric and transition metric, respectively. Following the notation in [27], the equation for state metric computation can be rewritten in the form of matrix-vector multiplication (Eq. (6)).

$$\begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix}_{k+1} = \begin{pmatrix} \lambda_{00} & \lambda_{01} & \lambda_{02} & \lambda_{03} \\ \lambda_{10} & \lambda_{11} & \lambda_{12} & \lambda_{13} \\ \lambda_{20} & \lambda_{21} & \lambda_{22} & \lambda_{23} \\ \lambda_{30} & \lambda_{31} & \lambda_{32} & \lambda_{33} \end{pmatrix}_k \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{pmatrix}_k \quad (6)$$

We should do the calculations in Eq. (6) like a normal matrix-vector multiplication but bearing in mind that multiplications and additions should be replaced by additions and minimum operations respectively. The arguments of the minimum operations are also kept in the survivor transition vector (STV).

$$STV_k = \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}_{k-1} \quad (7)$$

The elements of this vector, at time instant  $k$ , point to the previous symbols at time instant  $k - 1$ , and are stored in SMU.

#### F. CONVENTIONAL MLSE IMPLEMENTATION

In this section, we briefly review the implementation of conventional MLSE for 4-PAM,  $1 + D$  signaling. In ADC-based wireline receivers, the symbol rate is often much more than the maximum clock frequency that the digital processor can operate at. For this reason, digital operations are performed in parallel domain, often using a pre-defined serial-to-parallel conversion ratio. We use the lookahead method as explained in [27] to efficiently implement the MLSE. Besides, the transition and state metrics are computed based on carry-save representation in order to speed-up the operations.

To compute the transition metrics, we first expand Eq. (4) as:

$$TM_{ij} = y^2 - 2x_{ij}y + x_{ij}^2 \quad (8)$$

We note that  $y^2$  is the common term across all the transitions and does not contribute to the final select operation. Thus, it can be excluded from the TM equation:

$$TM_{ij} = x_{ij}^2 - 2x_{ij}y \quad (9)$$

In Eq. (9),  $x_{ij}^2$  does not depend on the input signal, thus it can be precomputed and given to the TMU block as a constant. Besides, if we assume 4-PAM levels are equal to 0, 1, 2, 3, then the  $1 + D$  channel turns it into 7 levels of 0, 1, 2, 3, 4, 5, 6. Therefore, we have 7 distinct values for  $x_{ij}$ , and  $x_{ij}y$  can be easily computed by simple add and shift operations.

To update state metrics at time instant  $k + 2$ , we can use Eq. (10).

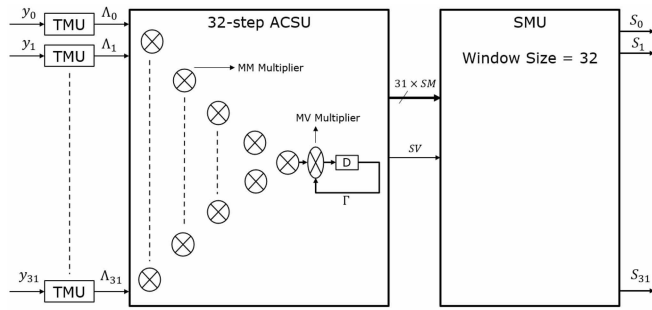
$$\Gamma_{k+2} = \Lambda_{k+1} \Gamma_{k+1} \quad (10)$$

where  $\Gamma$ , and  $\Lambda$  represent state vector and transition matrix respectively. This equation can be rewritten as

$$\Gamma_{k+2} = \Lambda_{k+1} \Lambda_k \Gamma_k = {}_2\Lambda_k \Gamma_k \quad (11)$$

Thus, we can first calculate  ${}_2\Lambda_k$  by doing matrix-matrix multiplication and then do the matrix-vector multiplication. The new transition matrix ( ${}_2\Lambda_k$ ) is called 2-step transition matrix, because it connects the state vectors which are 2 steps apart [27]. Moreover, the argument of the minimum operations done in computation of  ${}_2\Lambda_k$  should be kept in survivor transition matrix to be later used in the SMU logic for recovering the most probable transmitted sequence. This idea can be expanded to compute m-step transition matrix to update state metrics every m steps.

In this paper, we use a serial-to-parallel ratio of 32, so we calculate 32-step transition matrix in the form of a tree architecture and update state metrics every 32 steps (Fig. 11). This 32-step ACSU provides the SMU block with 31 survivor transition matrices and one survivor transition vector which can be used for data recovery. Moreover, to avoid overflow


**FIGURE 11.** Conventional MLSE for processing 32 parallel data.

in the computation of state metrics, we use a very simple normalization method explained in [28].

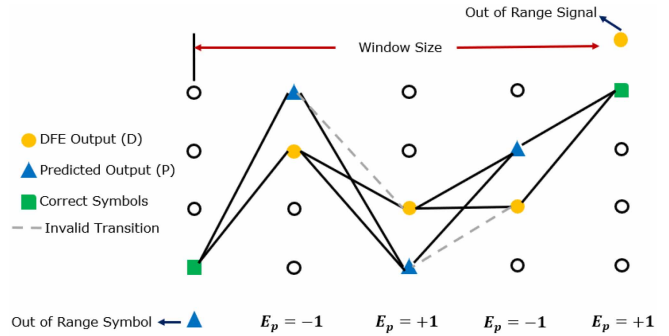
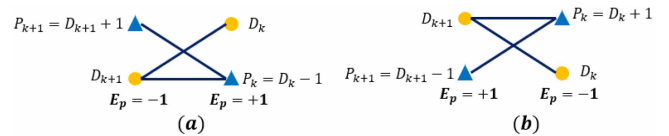
### III. PROPOSED ALGORITHM

As mentioned before, in this paper, the MLSE is used alongside a DFE. Therefore, we have access to the DFE output to guide the MLSE through the trellis diagram. Moreover, the MLSE is activated to correct the burst of errors which is known to most likely alternate between  $-1$  and  $+1$  until it hits an out-of-range status. Hence, we can simplify the algorithm with the following assumptions.

- In the trellis diagram, we move from right to left. Actually, the direction of movement can be chosen arbitrarily. However, Since the out-of-range signal triggers the MLSE engine, we choose that point as the initial point for the Viterbi algorithm.
- We assume that 4-PAM symbols take on 0, 1, 2, and 3 values. Besides, the channel is assumed to be  $1 + D$ .
- Error value alternates between  $-1$  and  $+1$  as we move through the symbols.
- There is one and only one error caused by noise within the MLSE window. That error indicates the beginning of the burst. The end of burst is where the out-of-range has been detected.

Before deploying the trellis diagram and applying Viterbi algorithm, we try to get as much information as we can from the DFE output and the out-of-range signal to more efficiently guide the MLSE engine.

Firstly, we know that the out-of-range signal indicates the end of burst, and the DFE output is most likely correct at that particular point. Besides, if the high-end out-of-range comparator is triggered, it implies that the hypothetical error at this point would have been  $+1$ . Thus, due to the alternating behaviour of the error, the previous symbol's error (previous in time) is  $-1$ , the second previous symbol's error is  $+1$ , and so on. Similarly, if the low-end out-of-range comparator is triggered, the previous symbol's error is  $+1$ , the second previous symbol's error is  $-1$ , and so on. Note that when we state the error is  $-1$  or  $+1$ , it does not necessarily mean that the DFE output is wrong at that point. It only means that if the DFE output is wrong, the correct symbol can be derived by subtracting that error from the DFE output. Therefore, we devise a block called symbol predictor (SP),


**FIGURE 12.** An example of trellis diagram in which the extra information is highlighted.

**FIGURE 13.** Trellis states and transitions when the current predicted error is (a)  $+1$ , (b)  $-1$ .

whose task is to compute the probable correct symbols using the error values and the DFE outputs within a fixed length of symbols (e.g., 32) ending in the out-of-range signal.

Secondly, as the SP moves backward (in time) through the symbols and computes the correct symbols based on the error pattern, it might output an invalid symbol (out of the range of the valid values). For instance, if the error value is  $+1$  at one particular point and the DFE output is 0, the predicted correct symbol is computed as  $-1$  which is an invalid value. A similar event happens if the error is  $-1$  and the DFE output is 3. In other words, SP might create a new out-of-range symbol, which implies the beginning of the burst must be somewhere between this new out-of-range signal and the original one detected by the DFE. Hence, the MLSE window can be adjusted dynamically and confined within these two points. If SP does not generate any invalid symbol, the window size is adjusted to its pre-defined maximum length (e.g., 32).

Finally, if the MLSE determines that the DFE output is correct at one particular point, the previous DFE output (previous in time) is also correct, because we have not yet reached the first error of the burst.

Fig. 12 illustrates an example of trellis diagram based on the above arguments. As we can see, in each step of the trellis diagram, there are only two valid states. One of them is the DFE output denoted by D and the other one is the SP output denoted by P. Besides, the dashed transitions in Fig. 12 are invalid, because they correspond to situations where the bursts of errors are terminated without encountering out-of-range signals. Therefore, the total number of transitions between two steps is reduced to three. To simplify the Viterbi algorithm we divide it into two separate cases. In one case, the algorithm describes a situation where the current error is  $+1$ , and in the other case  $-1$ . This is depicted in Fig. 13.

Since the channel is assumed to be  $1 + D$ , the expected value for each transition ( $x_{ij}$  in Eq. (9)) is the addition of the symbols connected by that transition. Hence, TM equations for Fig. 13a are

$$\begin{aligned} TM_{D_{k+1}, D_k} &= (D_k + D_{k+1})(D_k + D_{k+1} - 2y) \\ TM_{P_{k+1}, P_k} &= (D_k + D_{k+1})(D_k + D_{k+1} - 2y) \\ TM_{D_{k+1}, P_k} &= (D_k + D_{k+1} - 1)(D_k + D_{k+1} - 2y - 1) \end{aligned} \quad (12)$$

where  $D$ , and  $y$  are the values of the DFE and received level of the signal respectively.

Since we are interested in the relative difference between the transitions,  $TM_{P_{k+1}, P_k}$  is subtracted from all the transitions and the final result is divided by two. Hence:

$$\begin{aligned} TM_{D_{k+1}, D_k} &= 0 \\ TM_{P_{k+1}, P_k} &= 0 \\ TM_{D_{k+1}, P_k} &= y_k - (D_k + D_{k+1}) + 0.5 \end{aligned} \quad (13)$$

Similarly, for case b in Fig. 13, the transition metrics are calculated as:

$$\begin{aligned} TM_{D_{k+1}, D_k} &= 0 \\ TM_{P_{k+1}, P_k} &= 0 \\ TM_{D_{k+1}, P_k} &= (D_k + D_{k+1}) - y_k + 0.5 \end{aligned} \quad (14)$$

For the calculation of state metrics, we note that there is only one transition connecting a predicted symbol to the previous predicted symbol with a value of zero. Thus, the predicted symbol state metric remains zero throughout the MLSE window. However, the state metric of the DFE output is updated according to the following equations:

$$\begin{aligned} \text{CASE } a &\Rightarrow SM_{k+1} = \min(SM_k, y_k - (D_k + D_{k+1}) + 0.5) \\ \text{CASE } b &\Rightarrow SM_{k+1} = \min(SM_k, (D_k + D_{k+1}) - y_k + 0.5) \end{aligned} \quad (15)$$

According to these equations, if the minimum value is equal to the current state metric, the new state metric does not change and the survivor path connects two DFE outputs together. However, if the transition metric connecting DFE output to the predicted symbol is lower than the current state metric, the new state metric is updated with the new value and the survivor path switches to connect the DFE output to the predicted symbol. Therefore, if we apply this algorithm on all the symbols within the MLSE window, the final state metric will become equal to the global minimum of all the transitions metrics and at that point the DFE output is connected to the predicted symbol. Prior to that point in time, all the survivor paths go through the DFE outputs and after that point all the predicted symbols are connected together. In other words, the global minimum indicates the beginning of the burst. Thus, the algorithm can be simplified as below.

First, all the transition metrics are calculated according to the following equations:

$$\begin{aligned} \text{CASE } a &\Rightarrow TM_{D_{k+1}, P_k} = y_k - (D_k + D_{k+1}) \\ \text{CASE } b &\Rightarrow TM_{D_{k+1}, P_k} = (D_k + D_{k+1}) - y_k \end{aligned} \quad (16)$$

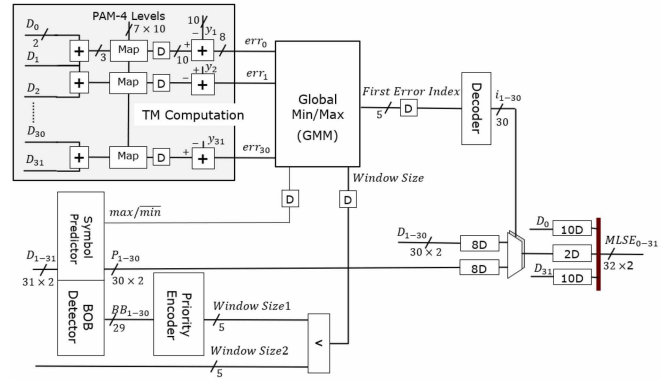


FIGURE 14. MLSE engine block diagram realizing reduced MLSE on demand.

Note that the constant 0.5 is removed from the equations, because it does not have any impact on the global minimum.

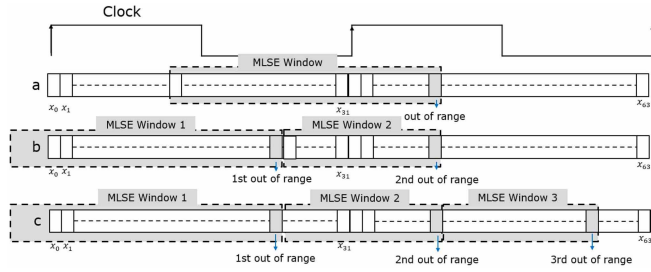
Second, the global minimum of all of these values within the MLSE window is derived. The symbol with the global minimum indicates the beginning of the burst. Hence, the DFE output is correct before that point (before in time) and all the next symbols should be replaced with the predicted symbols until the end of the burst.

## IV. PROPOSED IMPLEMENTATION OF THE REDUCED MLSE ON DEMAND

### A. $1 + D$ CASE

In this section, we explain how we can implement Reduced MLSE on Demand in hardware based on the algorithm discussed in Section III. Fig. 14 depicts the core of the algorithm. This block is designed for processing 32 parallel signals per each clock cycle. The main input signals are DFE outputs ( $D_0$ - $D_{31}$ ) and received signals after the channel ( $y_0$ - $y_{31}$ ). Besides, we assume that out-of-range signal has occurred at  $D_{31}/y_{31}$ . In other words, the data given to the MLSE engine is always aligned such that the end of burst is at the last index of the data ( $D_{31}$ ,  $y_{31}$ ). Later, we elaborate how this can be achieved given that the out-of-range signal can happen anywhere within the data block. Having DFE and channel outputs, we can compute transition metrics based on Eq. (16). To know which case of that equation should be used for each TM calculation, we should be aware of the error pattern. Since the error pattern is alternating, knowing the error at the point of out-of-range signal is sufficient to know the value of the error for the entire sequence. In this design, only for TM computations, we have assumed that the error is always  $-1$  at the point of out-of-range signal ( $D_{31}$ ). To consider the case when the error is actually  $+1$ , we look into Eq. (16), and realize that cases a and b are negative of each other. Thus, we should only negate all the computed TMs to have the correct values. Alternatively, instead of negation, we can take the maximum value as the start of the burst. In this paper, we chose the latter option.

In Fig. 14, the "Symbol Predictor" block computes the probable correct symbols based on the DFE output and the



**FIGURE 15.** Location of the out-of-range signal and the MLSE window in the data block for different clock situations (a) There is only one out-of-range signal in two successive clock cycles (b) There are two out-of-range signals in two successive clock cycles (c) There are three out-of-range signals in two successive clock cycles.

error value (Fig. 12). When an out-of-range symbol is generated due to these computations, the index of that symbol is provided by the “Beginning of Burst” (BOB) detector block. This index is used to limit the window size of the MLSE according to Fig. 12. Moreover, the window size could be also limited by another input called “Window Size 2” which is discussed later in this section. The “Symbol Predictor” also finds the value of the error at the point of the out-of-range signal and based on that notifies the “Global Min/Max” (GMM) block whether minimum or maximum operations should be performed. The GMM block finds the global minimum/maximum across all the TMs within the window. The index of the minimum/maximum value is the indicator of the start of the burst. Thus, all the symbols after this point are replaced with the predicted symbols.

As mentioned earlier, this MLSE engine works if the data is aligned such that the end of burst is at the end of the data block. However, we know normally the end of burst may happen anywhere within the data block (Fig. 15a). This alignment can simply be achieved by properly shifting the sequence which has partial overlap over two successive clock cycles. As a result, the output of the MLSE must also be shifted to align with the normal stream of data. These shifting operations are realized by “Barrel Shifter” blocks with amounts determined by the “End of Burst (EoB) Scanner” block as shown in Fig. 16.

The MLSE window size should be also limited here to avoid processing the already processed data. In Fig. 15a, there is only one out-of-range signal in two successive clock cycles. Thus, the MLSE window size can be set to its maximum value (32 in our implementation). However, if there are two out-of-range signals in two successive clock cycles (Fig. 15b), the MLSE window size may be required to be limited. Moreover, the MLSE engine can process just one out-of-range signal per each clock cycle. Therefore, if there are two out-of-range signals in the same clock cycle (Fig. 15c), we have to first split the data in two different clock cycles and then process them one by one. This splitting also necessitates merging the data back to one clock cycle after the above processing. The operations of determining the window size and splitting the data block is performed by the “Input Interface Controller” (IIC) block and the merging

operation is handled by the “Output Interface Controller” (OIC) block in Fig. 16. The MoD block outputs “Valid” signals as well as the recovered symbols. The “Valid” signal indicates whether or not the output of the MoD is valid. Each symbol whose corresponding valid signal is true should replace the DFE output at the end. As a last note, the designed MoD cannot handle more than three out-of-range signals in two successive clock cycles. This is to reduce the implementation complexity and its negligible effect on the overall error performance is confirmed by simulations.

Finally, note that the designed MoD is targeting a system structure similar to one shown in Fig. 6. Furthermore, the DFE designed in this structure is a  $1 + D$  DFE with the capability of out-of-range detection, and its implementation is done based on unrolling and pipelining techniques used in [13]. Fig. 17 shows more details about the implementation of the DFE.

### B. $1 + \alpha D$ CASE

So far, we have derived the equations and discussed the implementation of an on-demand MLSE for  $1 + D$  signaling. In this section, we explain what changes are required in the equations and the implementation to accommodate a more general case of  $1 + \alpha D$ .

First, as discussed in Section II-D, if  $\alpha \leq 0.5$ , a lot of EOBs are missed and the performance of the system approaches a DFE-based receiver. Therefore, our proposed receiver will lose its advantage if  $\alpha \leq 0.5$ .

Second, DFE implementation should be changed to cover the more general case of  $1 + \alpha D$ .

Third, the trellis diagrams would be identical with Fig. 12 and Fig. 13. However, transition metrics computation would be changed to:

Case a :

$$TM_{P_{k+1}, P_k} = 0$$

$$TM_{D_{k+1}, D_k} = (1 - \alpha)((\alpha D_{k+1} + D_k) - y_k) - \frac{(1 - \alpha)^2}{2}$$

$$TM_{D_{k+1}, P_k} = \alpha(y_k - (\alpha D_{k+1} + D_k)) + \alpha \left(1 - \frac{\alpha}{2}\right). \quad (17)$$

Case b :

$$TM_{P_{k+1}, P_k} = 0$$

$$TM_{D_{k+1}, D_k} = (1 - \alpha)(y_k - (\alpha D_{k+1} + D_k)) - \frac{(1 - \alpha)^2}{2}$$

$$TM_{D_{k+1}, P_k} = \alpha((\alpha D_{k+1} + D_k) - y_k) + \alpha \left(1 - \frac{\alpha}{2}\right) \quad (18)$$

By comparing these equations with Eq. (13) and Eq. (14), we realize that more computations are needed for  $TM_{D_{k+1}, P_k}$ , and  $TM_{D_{k+1}, D_k}$  is no longer zero for the general case.

Like the special case of  $1 + D$ , in the calculation of the state metrics, the predicted symbol state metric remains zero throughout the MLSE window. However, DFE output state metric is computed as:

$$SM_{k+1} = \min(SM_k + TM_{D_{k+1}, D_k}, TM_{D_{k+1}, P_k}) \quad (19)$$



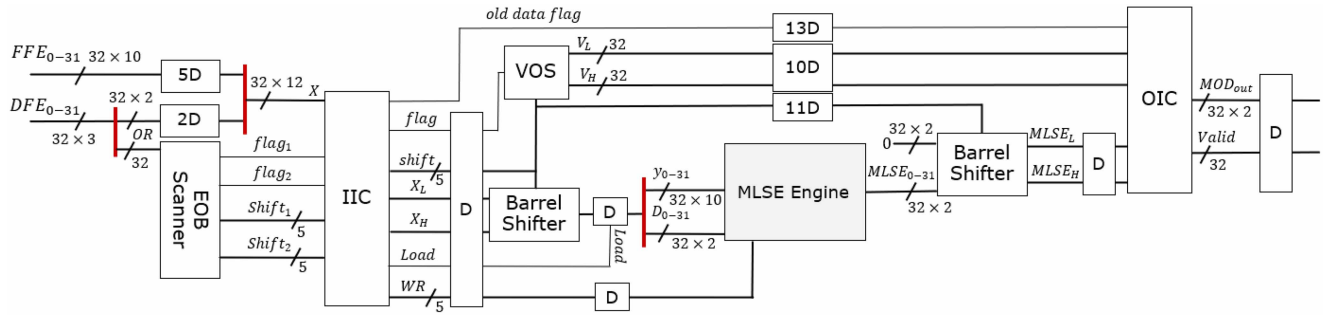


FIGURE 16. MoD block diagram.

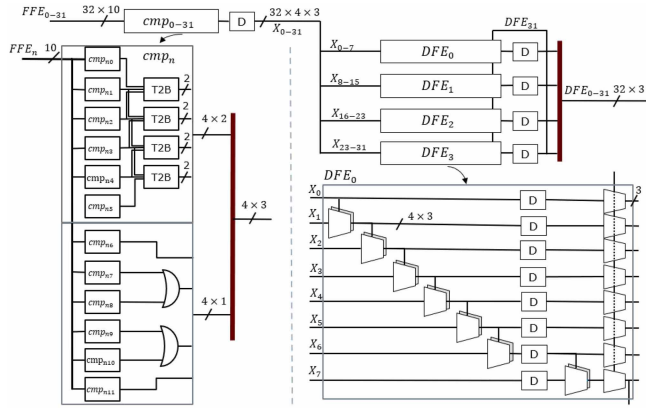


FIGURE 17. DFE block diagram.

If  $SM_k + TM_{D_{k+1}, D_k} < TM_{D_{k+1}, P_k}$ , two DFE outputs are connected together. Otherwise, the survivor path is the one connecting DFE with the predicted symbol. By recursively applying Eq. (19) on all the symbols within the MLSE window, the final value of the state metric would be equal to:

$$SM_n = \min(e_0, e_1, \dots, e_i, e_{n-1}) \quad (20)$$

where  $n$  is the size of the MLSE window, and  $e_i$  is equal to:

$$e_i = TM_{D_{i+1}, P_i} + \sum_{k=i+1}^{n-1} TM_{D_{k+1}, D_k}, \text{ for } i \neq n-1$$

$$e_{n-1} = TM_{D_n, P_{n-1}} \quad (21)$$

Like the particular case of  $1+D$ , the argument of the minimum operation indicates the beginning of the burst. However, in  $1+\alpha D$ , we should first compute the  $e_i$ s based on Eq. (21) and then apply the minimum operation. It can be shown that we need  $2n$  adders for these extra computations.

In summary, the MLSE engine shown in Fig. 14 requires two modifications in order to accommodate the  $1+\alpha D$  case. First, the TM computation section should be modified according to Eq. (17) and Eq. (18). Second,  $2n$  additional adders must be inserted between TM computation block and the GMM. These modifications are expected to increase the receiver complexity compared to the  $1+D$  case, but will remain competitive with respect to the conventional MLSE implementation.

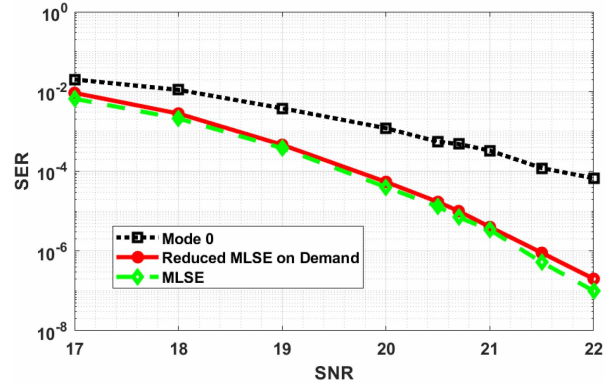


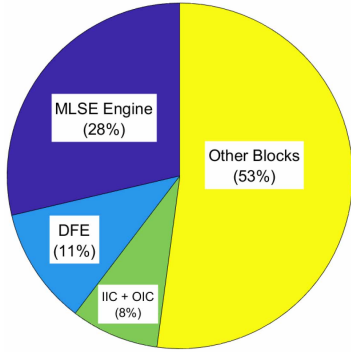
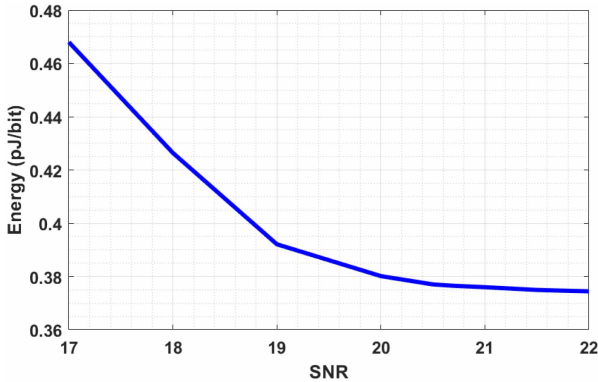
FIGURE 18. SER vs SNR for mode-0 of the error correction, reduced MLSE-on-Demand, and conventional MLSE.

## V. SIMULATION RESULTS AND ANALYSIS

We have designed and synthesized both the conventional MLSE and our proposed MLSE-on-Demand in TSMC 16nm FinFet process. As explained earlier, the data is assumed to be 4-PAM and passed through a  $1+D$  channel. It is also de-muxed into 32 parallel lines, which is a typical serial-to-parallel ratio in high speed wireline transceiver implementations. The RTL simulation results are depicted in Fig. 18. The square-marked curve shows the symbol error rate (SER) vs. SNR for mode-0 of the error correction (DFE only). The other two curves show the performance of the conventional MLSE and the proposed MLSE-on-Demand. As we can see, there is a negligible difference between these two algorithms, even though the proposed MLSE-on-Demand is much less complex than the conventional algorithm. To quantify the advantage, we can refer to Table 1. The area occupied by the proposed algorithm (including everything shown on Fig. 6) is almost 15 times less than the area of conventional MLSE and its power consumption is reduced by 10 times. This power reduction is a result of simplifications in the algorithm and does not yet include further reduction when the MLSE function is invoked on an on-demand basis. According to Fig. 19, 28% of the total power is consumed by the MLSE engine and the rest is consumed by the other blocks including DFE, Decoder, etc. If we turn off the MLSE engine when there is no error, the power is reduced by 28% during that time. Therefore, the average power consumption depends on the SNR as shown by Fig. 20. As SNR becomes

**TABLE 1.** Comparison between conventional MLSE and reduced MLSE-on-demand synthesized in 16nm FinFET TSMC technology.

	MLSE	RMoD
Max Frequency (GHz)	1GHz	1.5GHz
Latency (Clock cycles)	21	22
Area ( $mm^2$ )	0.230	0.015
Energy Efficiency ( $\mu J/b$ )	5.3	0.52

**FIGURE 19.** Power consumption of the MLSE engine compared to the other blocks.**FIGURE 20.** Energy Efficiency vs SNR.

higher, SER becomes lower, and the power consumption reduces. The minimum power consumption is 72% of the number shown in Table 1 and is achieved when the SER approaches zero.

The comparison between the proposed algorithm and some other implemented algorithms is summarized in Table 2. To quantitatively compare our proposed algorithm with the other state-of-the-art designs, we define figure of merit (FoM) as follow:

$$FoM = \frac{N}{(A/size^2) \times (power/DR)} \quad (22)$$

where  $N$  is the number of transitions between two successive Viterbi steps,  $A$  is the area of the implemented algorithm,  $size$  is the feature size of the technology,  $power$  is the power consumption, and  $DR$  is the maximum data rate. Note that the  $power/DR$  term represents the energy efficiency expressed in  $J/bit$ .

Since a more complex algorithm occupies more area, FoM should be inversely proportional to the area. However, to have a fair comparison, we normalize the area to the square of the feature size as expressed in Eq. (22). Energy efficiency

is the other important factor that is considered in the FoM computation. Moreover, we should also consider the computational complexity of a conventional trellis diagram sketched for the MLSE. The computational complexity is roughly proportional to the number of transitions existing between two successive Viterbi steps. As the number of transitions increases, the number of logic blocks used for the MLSE implementation increases accordingly. This is also reflected in the FoM. As an example, the FoM for [26] in Table 2 is computed as follow:

$$FoM = \frac{4^4}{\frac{3.6 \times 10^5}{(14 \times 10^{-3})^2} \times 4.1 \times 10^{-12}} = 3.4 \times 10^4 \quad (23)$$

Table 2 also shows FoM calculated for each case, and demonstrates the efficiency advantage of our proposed reduced MLSE on demand (RMoD).

## VI. CONCLUSION

In this paper, we examined different ways to cancel the propagated errors caused by a DFE. This is particularly useful when the DFE is customized for equalizing a  $1 + D$  channel, where the probability of error propagation reaches its maximum value and cancelling the propagated errors becomes a necessity. If we also use the ISI-induced correlation between consecutive symbols, even the first error which has triggered the propagation might be corrected. To take the correlation into account we use MLSE, but unlike the conventional one, we use it on demand once DFE alerts that an error has occurred. The whole procedure of correcting the entire burst of errors can be summarized as follows.

First, end of burst is detected by out of range signal detectors. Second, the start of the burst is detected by the MLSE. Our proposed design demonstrates that there is no traceback required if our goal is to detect the start of the burst. Finally, knowing the error pattern helps us correct all the symbols in the burst whose start and end are detected by MLSE and the out-of-range signal detector respectively.

To verify the performance of the receiver, both a conventional and the proposed MLSE have been designed and synthesized in 16nm FinFET TSMC Technology for the special case of 4-PAM signaling and a  $1 + D$  channel. The simulation results demonstrates that the MLSE-on-Demand can process upto 96Gb/s while consuming 10 times less power and occupying 15 times less area than the conventional MLSE. Data rate can be increased by increasing the number of parallel branches, and the power consumption can be further decreased if we only turn on the MLSE upon detection of an error (a truly on-demand operation). Moreover, this proposed idea can be expanded to cover a more general case of  $1 + \alpha D$  or higher number of taps.

## ACKNOWLEDGMENT

The authors thank Huawei Canada for their expertise and assistance throughout the course of this research. Access to CAD tools was provided by CMC Microsystems. They also thank NSERC of Canada for partial funding of this research.

**TABLE 2. Performance comparison with other implementations.**

	[29]	[24]	[26]	This work (MLSE)	This work (RMOd)
Technology	90nm	65nm	14nm	16nm	<b>16nm</b>
Area ( $mm^2$ )	5.7	0.21	0.36	0.23	<b>0.015</b>
Energy (pJ/b)	74.7	3.9	4.1	5.3	<b>0.52</b>
Data Rate (Gb/s)	32	10	25.6	64	<b>96</b>
Modulation	2-PAM	2-PAM	4-PAM	4-PAM	<b>4-PAM</b>
EQ Structure	3-tap MLSE	3-tap MLSE	4-tap MLSE	2-tap MLSE (1 + D)	<b>2-tap MLSE (1 + D)</b>
FoM (Eq. 22)	$1.5 \times 10^2$	$4.1 \times 10^4$	$3.4 \times 10^4$	$3.4 \times 10^3$	$5.3 \times 10^5$

## REFERENCES

- [1] S. Palermo, S. Hoyos, S. Cai, S. Kiran, and Y. Zhu, "Analog-to-digital converter-based serial links: An overview," *IEEE Solid-State Circuits Mag.*, vol. 10, no. 3, pp. 35–47, Aug. 2018.
- [2] S. Kiran, S. Cai, Y. Luo, S. Hoyos, and S. Palermo, "A 52-gb/s ADC-based PAM-4 receiver with comparator-assisted 2-bit/stage SAR ADC and partially unrolled DFE in 65-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 3, pp. 659–671, Mar. 2019.
- [3] Aurangozeb, A. D. Hossain, M. Mohammad, and M. Hossain, "Channel-adaptive ADC and TDC for 28 gb/s PAM-4 digital receiver," *IEEE J. Solid-State Circuits*, vol. 53, no. 3, pp. 772–788, Mar. 2018.
- [4] J. Im *et al.*, "6.1 a 112Gb/s PAM-4 long-reach Wireline transceiver using a 36-way time-interleaved SAR-ADC and inverter-based RX analog front-end in 7nm FinFET," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2020, pp. 116–118.
- [5] T. Ali *et al.*, "6.2 a 460mW 112Gb/s DSP-based transceiver with 38dB loss compensation for next-generation data centers in 7nm FinFET technology," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2020, pp. 118–120.
- [6] M. A. LaCroix *et al.*, "8.4 a 116Gb/s DSP-based Wireline transceiver in 7nm CMOS achieving 6pJ/b at 45dB loss in PAM-4/duo-PAM-4 and 52dB in PAM-2," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2021, pp. 132–134.
- [7] D. Xu *et al.*, "8.5 a scalable adaptive ADC/DSP-based 1.25-to-56Gbps/112Gbps high-speed transceiver architecture using decision-directed MMSE CDR in 16nm and 7nm," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2021, pp. 134–136.
- [8] P. Mishra *et al.*, "8.7 a 112Gb/s ADC-DSP-based PAM-4 transceiver for long-reach applications with >40dB channel loss in 7nm FinFET," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2021, pp. 138–140.
- [9] M. Pisati *et al.*, "A 243-mW 1.25–56-gb/s continuous range PAM-4 42.5-dB IL ADC/DAC-based transceiver in 7-nm FinFET," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 6–18, Jan. 2020.
- [10] B. Yoo *et al.*, "6.4 a 56Gb/s 7.7mW/gb/s PAM-4 Wireline transceiver in 10nm FinFET using MM-CDR-based ADC timing skew control and low-power DSP with approximate multiplier," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2020, pp. 122–124.
- [11] K. Parhi, "Design of Multigigabit multiplexer-loop-based decision feedback Equalizers," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 4, pp. 489–493, Apr. 2005.
- [12] D. Oh and K. Parhi, "Low complexity design of high speed parallel decision feedback Equalizers," in *Proc. Appl. Spec. Syst. Archit. Process. (ASAP) IEEE Conf.*, 2006, pp. 118–124.
- [13] T. Shibusaki *et al.*, "A 56-gb/s receiver front-end with a CTLE and 1-tap DFE in 20-nm CMOS," in *VLSI Circuits Dig. Tech. Papers Symp.*, 2014, pp. 1–2.
- [14] A. Cervero *et al.*, "6.1 a 100Gb/s 1.1pJ/b PAM-4 RX with dual-mode 1-tap PAM-4 / 3-tap NRZ speculative DFE in 14nm CMOS FinFET," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 2019, pp. 112–114.
- [15] S. Kiran, S. Cai, Y. Zhu, S. Hoyos, and S. Palermo, "Digital Equalization with ADC-based receivers: Two important roles played by digital signal Processing in designing analog-to-digital-converter-based Wireline communication receivers," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 62–79, May 2019.
- [16] J. Bailey *et al.*, "8.8 a 112Gb/s PAM-4 low-power 9-tap sliding-block DFE in a 7nm FinFET wireline receiver," in *Proc. IEEE Int. Solid State Circuits Conf. (ISSCC)*, 2021, pp. 140–142.
- [17] M. Yang, S. Shahramian, H. Shakiba, H. Wong, P. Krotnev, and A. C. Carusone, "Statistical BER analysis of Wireline links with non-binary linear block codes subject to DFE error propagation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 1, pp. 284–297, Jan. 2020.
- [18] Y. C. Lu, H. Wang, D. Tonietto, and D.-J. Zang, "DFE error propagation characteristics in real 56Gbps PAM4 high-speed links with pre-coding and impact on the FEC performance," in *Proc. DesignCon*, 2017, pp. 9–10.
- [19] Y. C. Lu, P.-C. Zhao, W.-Y. Wang, Z.-L. Huang, H. Wong, and D. Tonietto, "A comparative study of Equalization schemes for 112G PAM4 links," in *Proc. DesignCon*, 2019, pp. 5–7.
- [20] G. D. Forney, "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.
- [21] 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.
- [22] H.-M. Bae, J. B. Ashbrook, J. Park, N. R. Shanbhag, A. C. Singer, and S. Chopra, "An MLSE receiver for electronic dispersion compensation of OC-192 fiber links," *IEEE J. Solid-State Circuits*, vol. 41, no. 11, pp. 2541–2554, Nov. 2006.
- [23] G. Fettweis and H. Meyr, "High-speed parallel Viterbi decoding: Algorithm and VLSI-architecture," *IEEE Commun. Mag.*, vol. 29, no. 5, pp. 46–55, May 1991.
- [24] S. Song, K. D. Choo, T. Chen, S. Jang, M. P. Flynn, and Z. Zhang, "A maximum-likelihood sequence detection powered ADC-based serial link," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 7, pp. 2269–2278, Jul. 2018.
- [25] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 13–20, Jan. 1988.
- [26] 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. ESSCIRC Conf. 42nd Eur. Solid-State Circuits Conf.*, Lausanne, Switzerland, 2016, pp. 309–312.
- [27] G. Fettweis and H. Meyr, "High-rate Viterbi processor: A systolic array solution," *IEEE J. Sel. Areas Commun.*, vol. 8, no. 8, pp. 1520–1534, Oct. 1990.
- [28] G. Fettweis and H. Meyr, "A 100 Mbit/s Viterbi decoder chip: Novel architecture and its realization," in *Proc. IEEE Int. Conf. Commun.*, Apr. 1990, pp. 463–467.
- [29] T. Veigel, T. Alpert, F. Lang, M. Grozing, and M. Berroth, "A Viterbi equalizer chip for 40 gb/s optical communication links," in *Proc. Eur. Microw. Integr. Circuits Conf.*, 2013, pp. 49–52.



**MOHAMMAD EMAMI MEYBODI** (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Amirkabir University of Technology (Tehran Polytechnic), Iran, in 2014, and the M.Sc. degree in electrical engineering from the University of Tehran, Iran, in 2017. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Toronto, Canada. He was collaborating with the Research Center, Amirkabir University of Technology from

2014 to 2018. His current research interests include high speed communication, RF communication, visible light communication, and analog and digital circuit design.



**HECTOR GOMEZ** (Member, IEEE) received the B.S.E.E. degree from the Universidad Industrial de Santander, Bucaramanga, Colombia, in 2009, the M.S.E. degree from the Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico, in 2011, and the Ph.D. degree from the Universidad Industrial de Santander in 2019. From 2012 to 2015, he was an Assistant Professor with UNISANGIL, San Gil, Colombia. He is currently a Postdoctoral Fellow with the University of Toronto, Canada. He has authored and coauthored over 20 conference/journal publications. He holds a patent and has one pending patent in the area of integrated circuits. His research interests include low power clock generation, hardware for security, and high-speed interfaces. He has served on technical program committees for IEEE CAS conferences. He is currently serves as a Technical Program Committee Member of the IEEE Latin American Symposium on Circuits and Systems.

He has served on technical program committees for IEEE CAS conferences. He is currently serves as a Technical Program Committee Member of the IEEE Latin American Symposium on Circuits and Systems.



**HOSSEIN SHAKIBA** (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran, in 1985 and 1989, respectively, and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Toronto, Canada, in 1997. He has over 35 years of teaching, research, design, and management experience in the area of analog circuit and system design for various

applications with focus on wireline communication in both the industry and academia. He is currently working on system and circuit development for next generation serial links at Huawei Canada in collaboration with the wireline industry with emphasis on link design, modeling, and analysis including statistical and signal integrity. He is also actively involved in conducting research with various universities and co-supervises several graduate students.



**YU-CHUN LU** (Member, IEEE) received the B.E. degree in communication and information systems from the School of Electronics and Information Engineering, Beijing Jiaotong University (BJTU), Beijing, China, in 2005, and the Ph.D. degree in communication and information systems from the Institute of Lightwave Technology, BJTU in 2010. From January 2007 to August 2008, he was a Visiting Researcher with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada. In 2010, he

joint Huawei Technologies, where he is currently the Engineer of Research and a Team Leader. His current interests include equalization, CDR, FEC, modeling and simulation methodology of high speed optical and electrical links. He is currently working on 224Gbps and beyond technologies for optical and electrical links and next generation Ethernet.



**ALI SHEIKHOESLAMI** (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Shiraz University, Iran, in 1990, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Toronto, Canada, in 1994 and 1999, respectively. In 1999, he joined the Department of Electrical and Computer Engineering, University of Toronto, where he is currently working as a Professor. He was on research sabbatical with Fujitsu Labs in 2005–2006, and Analog Devices in 2012–2013. He has

coauthored more than 100 journal, magazine, and conference papers, ten patents, and a graduate-level textbook titled *Understanding Jitter and Phase Noise—A Circuits and Systems Perspective*. His research interests include analog and digital integrated circuits, high-speed signaling, VLSI memory design, and CMOS annealing. He was a coauthored the CICC2017 Outstanding Student Paper Award. He served on the Memory, Technology Directions, and Wireline Subcommittees of the International Solid-State Circuits Conference (SSCS), in 2001–2004, 2002–2005, and 2007–2013, respectively. He currently serves as the Education Chair for ISSCC and SSCS. He is a Distinguished Lecturer (DL) of the Society and oversees its DL and webinar programs. He is an Associate Editor for the *Solid-State Circuits Magazine*, in which he has a regular column entitled *Circuit Intuitions*. He was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS from 2010 to 2012, and the Program Chair for the 2004 IEEE ISMVL. He has received numerous teaching awards including the 2005 and 2006 Early Career Teaching Award and the 2010 Faculty Teaching Award, from the Faculty of Applied Science and Engineering, University of Toronto. He is a Registered Professional Engineer in Ontario, Canada.