High-Performance VLSI Architecture of Decision Feedback Equalizer for Gigabit Systems

Chih-Hsiu Lin, An-Yeu (Andy) Wu, and Fan-Min Li

Abstract—This brief addresses the design of a *decision feedback equalizer* (DFE) for gigabit throughput rate. It is well known that the feedback loop in a DFE limits an upper bound of the achievable speed. For a L -tap *feedbackward filter* (FBF) and M -pulse *amplitude modulation*, Parhi (1991) and Kasturia and Winters (1991) reformulated the FBF as a $(M)^L$ -to-1 multiplexer. Due to the reformulation, the overhead of extra adders and extra multiplexers are as large as $(M)^L$. The required hardware overhead should be more severe when the DFE is implemented in parallel. In this brief, we propose two new approaches to implement the DFE when gigabit throughput rate is desired. The first approach is *partial pre-computation* scheme, which can trade-off between hardware complexity and computational speed. The second approach is *two-stage pre-computation* scheme, which can be applied to higher speed applications. In the latter case, we can reduce the hardware overhead to about $2(M)^{-(L/2)}$ times of [1], [2], and the iteration bound is $(\log_2 W + 2)/(L/2 + 1) + (\log_2 M)$ multiplexer-delays, where W is the wordlength of weight coefficient of a FBF. We demonstrate the proposed architectures by apply it to the 10 Gbase-LX4 optical communication systems.

Index Terms—Decision feedback equalizer (DFE), gigabit system, partial pre-computation scheme, two-stage pre-computation scheme.

I. INTRODUCTION

PIPELINING scheme is successfully employed to increase the computational speed in the communication equalizer designs [3], [4]. Another approach to achieve high-speed computation is to implement an equalizer in parallel [5], [6]. Exploiting both pipelining and parallel processing for high-speed applications is straightforward for non-recursive computations. However, recursive computations, such as *decision feedback equalizers (DFEs)*, can not be easily pipelined or processed in parallel due to the feedback loops in these filters. For a filter with a loop, the retiming approach [7] can be used to move the delay elements from a shorter path to a longer path in loops. Then, a smaller critical path can be obtained. However, the retiming approach can not achieve the iteration bound in most cases. In order to achieve iteration bound, we can unroll a loop, which is referred to as the *unfolding scheme* [4], [6], and then apply the retiming approach to the unfolded VLSI architecture.

Fig. 1 shows a conventional DFE architecture. The critical path of this DFE is one multiplier, one slicer, and two adders as

Manuscript received December 7, 2004; revised July 7, 2005. This work was supported in part by MediaTek Incorporation and the National Science Council, R.O.C., under Grant NSC 92-2220-E-002-012. This paper was recommended by Associate Editor J. Liu.

The authors are with Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, Taiwan 106, R.O.C. (e-mail: andywu@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCSII.2006.881165

Fig. 1. Architecture of a DFE and its iteration bound.

drawn in bold line. For 10 Gbase-LX4 optical communication systems [8], the modulation scheme is 2 *pulse amplitude modulation* (PAM); therefore, the multiplier can be replaced by one 2-to-1 multiplexer. The iteration bound reduces to one multiplexer and two adders. These two adders can be implemented in a full adders and a 2-input vector merging. Take the wordlength of weight coefficient in the *feedbackward filter* (FBF) $W = 8$ for example, the latency of the 2-input vector merging can be as short as 5 multiplexer-delays. Therefore, the critical path of this DFE is about 1.1 ns in UMC 0.18- μm cell libraries. It does not meet the required speed, i.e., 0.32 ns. The authors of [1] and [2] reformulate the FBF as 2^L -to-1 multiplexers. The iteration bound is reduced to one multiplexer. The delay time of one multiplexer is and a margin for a delay element 0.14 and 0.25 ns, respectively, in UMC 0.18- μm cell libraries. Hence, the computation speed of the architecture [1], [2] still cannot provide a delay element with enough margins. In addition, although the unfolding approach can be employed in [1], [2] to achieve the desired throughput rate, the overhead will be extremely large.

Motivated by [1] and [2], two new schemes are proposed in this brief. The first scheme is to pre-compute and sum the partial outputs of the FBF. This approach can be used to trade off between hardware complexity and computational speed. For higher speed applications, the second approach is proposed to reformulate the FBF as two-stage pre-computation. For M -PAM modulations and a L -tap FBF with wordlength W , we can reduce the hardware overhead to about $2(M)^{(-L/2)}$ times of [1], [2]. The iteration bound can be as low as $(\log_2 \mathbf{W} + 2)/(L/2 + 1) + (\log_2 \mathbf{M})$ multiplexer-delays.

The rest of this brief is organized as follows. In Section II, we review the reformulated FBF scheme. We propose the two new modified approaches in Section III. In Section IV, we apply the proposed two-stage pre-computation approach to 10 Gbase-LX4 optical communication systems. Finally, we conclude our work in Section V.

II. REVIEW OF THE REFORMULATED FBF SCHEME [1], [2]

We show a 2-tap FBF of a DFE in Fig. 2(a). The modulation scheme is 2-PAM. Fig. 2(b) is an alternative architecture by using retiming scheme. The critical path is one multiplexer

Fig. 2. (a) Reformulated 2-tap FBF of a DFE [1], [2]. (b) Retimed implementation of the reformulated 2-tap FBF.

delay. Furthermore, for an L -tap FBF, the fanout of the last stage multiplexer is $2^L - 1$. This will increase the delay time of this multiplexer. By using this reformulated approach, the whole overhead cost becomes 2^L adders and $(2^L - 1)$ 2-to-1 multiplexers. However, this architecture cannot provide margin enough for a delay element to process correctly for a 3.125 Gb/s throughput rate in 10 Gbase-LX4 optical communication systems. To achieve 3.125 Gb/s, unfolding approach must be adopted. For an N -unfolding implementation, the hardware cost is about N times. However, this hardware cost of this architecture is extremely large, especially when L is large. Parhi proposed a look-ahead scheme for multiplexers in [10] to solve this problem.

III. PROPOSED PRE-COMPUTATION APPROACHES

In Fig. 1, when the transmitted signal is modulated by 2-PAM, the iteration bound is the computation delay of 2 adders and 1 multiplexer. The delay of a full adder and multiplexer is 0.28 and 0.14 ns in UMC 0.18- μm cell libraries, respectively. Therefore, the conventional DFE cannot be employed to the communication systems, in which the throughput rate is higher than gigabit per second. First, we discuss the partial pre-computation approach. Next, we present the two-stage pre-computation approach.

A. Partial Pre-Computation Scheme

Motivated by [1] and [2], we pre-compute and sum up the partial outputs of the FBF, and then use a multiplexer to select the correct output. By doing so, extra delay elements can be used to pipeline the feedback loops. To clarify the presentation, we denote T_M , T_{add} , T_{FA} , and T_{mux} as the delay of one multiplier, adder, full adder, and multiplexer, respectively. As depicted in Fig. 1, the critical path is in the inner loop, i.e., the path through multiplier a_1 . When the transmitted signal is modulated as M -PAM, the multiplier a_1 can be replaced by $\log_2 M$ -to-1 multiplexer. Furthermore, we can implement the adders by using carry-save adder, and then apply a multiplexer-based tree binary-look ahead carry generation adder [9] to generate the output. For a wordlength W adder, the latency of a two-binary-adder by using [9] is $(\log_2 W + 2)T_{\text{mux}}$. Therefore, the whole delay time can be reduced to $2T_{\text{FA}} + (\log_2 W + \log_2 M + 2)T_{\text{mux}}$.

Fig. 3. Architecture of a DFE by using 2-terms pre-computation architecture.

Fig. 4. Architecture of a DFE by using N -terms pre-computation approach.

To speed up the computation of a DFE, we must reduce the critical path in a FBF. If we pre-compute N terms of a FBF, then we have extra N -delay elements to pipeline the feedback loop. An example of 2-terms partial pre-computation is depicted in Fig. 3. The critical path can be reduced to $1/3$ times. Although, the critical path is reduced to $1/(N+1)$ ($N = 2$ for 2-terms partial pre-computation) times in the inner loop, the critical path is bounded by the adders in the rest of the FBF. Implementing a DFE of Fig. 3, we can use the N delay elements to pipeline the inner loop, and the critical path of the implementation can be reduced to T_{FA} , i.e., 0.28 ns in UMC 0.18- μm cell libraries. Therefore, the throughput rate of the partial pre-computation can be up to gigabit. Moreover, we can implement a FBF in an alternative way as shown in Fig. 4 when N is large enough. We use a summation tree to reduce to critical path in the FBF.

In summary, for a small N -term pre-computation, the critical path is about $(2T_{\text{FA}} + (\log_2 W + \log_2 M + 2)T_{\text{mux}})/(N+1)$. We implement a DFE as shown in Fig. 3. For a larger N , we implement a FBF as demonstrated in Fig. 4 to achieve a shorter critical path. The extra hardware costs of both architectures are 2^N adders and 2^N 2-to-1 multiplexers in N -terms pre-computation DFE. In general, N must be chosen as smaller than $L/2$. This will be obvious in next section.

B. Two-Stage Pre-Computation Scheme

In partial pre-computation method, we can trade-off between hardware cost and computing speed. However, for very high-

Fig. 5. Architecture of an L -tap FBF by applying two-stage pre-computation approach.

speed applications, such as 10 Gbase-LX4 optical communication systems, a small N -terms pre-computation can not provide a delay element with margin enough. In current UMC 0.18- μm cell libraries. As shown in Fig. 4, the critical path is limited by computing the sum of the outputs of $a_{N+1}, \dots$, and a_L . Therefore, if these terms are also pre-computed separately and simultaneously, we can shorten the path delay just at the price of smaller hardware cost. As illustrated in Fig. 5, the L -tap FBF is divided into two parts: One is the summation of the last terms, i.e., weight a_{I+1} and a_L , called the first stage pre-computation. The second stage sums the first I terms. The output, $y(n)$, of the FFF can be added by one of these two parts. Below, we discuss the properties of this scheme.

- 1) *Partition scheme*: The outputs of each tap-delay line in the FBF can be partitioned into one of two parts, but leads to different iteration bound. The best grouping is to put the first terms in one group, and the rests are grouped together.
- 2) *Asymptotic iteration bound*: For a L -tap FBF, the iteration bound of the two-stage pre-computation method is $1/(I+1)T_{\text{add}} + (\log_2 M)T_{\text{mux}}$. Since the input number of this adder is 2, we can implement it by using a multiplexer-based tree binary-look ahead carry generation Adder [9]. The latency of a W -bit binary adder can be as short as $(\log_2 W + 2)T_{\text{mux}}$. The complexity of the proposed architecture is as only $2(M)^{(-L/2)}$ time of [1], [2]. Moreover, the iteration bound of both two architectures is close to $(\log_2 M)T_{\text{mux}}$ [1], [2] as L becomes large.
- 3) *Low power*: For a communication system in which the channel changes very slowly, we can compute these sums of each stage and store in memories. In the first stage pre-computation, no new data is inputted when a slicer operates in data mode. We can compute the sum, store in memories. We need not update at each symbol time; therefore, the computing power can be saved. In the second stage, $y(n)$, the output of FFF, changes in each sample time, either when a slicer operates in training mode or data mode. Therefore, the computing power can not be saved in the second stage.
- 4) *Large fanout impairment*: As shown in Fig. 2(b), the fanout number of the multiplexer in the last stage is exponential

TABLE I
HARDWARE COMPLEXITY AND CRITICAL PATH IN FBF. THE NUMBER AND THE WORD-LENGTH OF FBF IS L (L IS ASSUMED EVEN) AND W , RESPECTIVELY. THE MODULATION SCHEME IS M -PAM

Cell name	$A_{02}, A_{12}, A_{22}, A_{32}$	$B_{02}, B_{12}, B_{22}, B_{32}$	$B_{01}, B_{11}, B_{21}, B_{31}$	The rest inverted multiplexers
Type	X4	X4	X2	X1

Fig. 6. Block diagram of equalizers in a receiver. FFF and FBF denote feed-forward filter and FBF, respectively.

to the tap-size of the FBF. This will increase the delay time of the last multiplexer. In the proposed architecture, the fanout number is degraded extremely compared with [1], [2], so that the increased latency can be mitigated. However, each output of the first stage is carried into $(M)^I$ adders, and the fanout of first stage is equal to wordlength of the FBF. The large fanout will also increase the latency. In the practical design, some approaches can be used to mitigate the impairment. This will be discussed in Section IV.

We show the comparison of critical path and hardware complexity in Table I.

IV. APPLICATION TO 10 GBASE-LX4 OPTICAL COMMUNICATION SYSTEMS

In 10 Gbase-LX4 optical communication systems, the communication is over 4 fibers (see Fig. 6). The transmitted signal is coded by 8 B/10 B scheme and modulated by 2-PAM. The actual data rate in each path is 3.125 Gb/s. That is, the clock time of an equalizer must be less than 0.32 ns. The delay of a full adder and a delay element in UMC 0.18- μm cell libraries is 0.28 ns and 0.25 ns, respectively. The designed tap-size of the FFF and FBF is 8 and 6, respectively, and the wordlength of weight coefficient in a FBF is 8.

In this application, we use the two-stage pre-computation scheme to design a FBF. To implement this FBF with the lowest hardware cost, I must be chosen as 3. Then, the iteration bound is $2.25 T_{\text{mux}}$ and the cost is 16 adders and 14 multiplexers. As shown in Fig. 6, we design the DFE by using 4-unfolding approach to provide sufficient margin for a delay element.

Fig. 7. Block diagram of the 4-unfolding architecture of the FBF. A and B denote the 2^3 -to-1 multiplexers, and C denotes the vector merge unit and slicer operation.

A. Architecture of a 6-Tap FBF With 4-Unfolding

The block diagram of the 4-unfolding architecture is illustrated in Fig. 7. The symbol $A_0 \sim A_3$ and $B_0 \sim B_3$ denote the operation of the 2^3 -to-1 multiplexers. Based on the previous outputs, $A_0 \sim A_3$ and $B_0 \sim B_3$ choose one of the 8 combinations in Stage 1 and in Stage 2, respectively. The symbol $C_0 \sim C_3$ denotes the operation of the Vector Merge Unit and Slicer. According to Fig. 7, we depict the detail of the 4-unfolding architecture of a 6-tap FBF in Fig. 8. The delay time of a *vector merge* (VM) Unit is $5T_{MUX}$ [9]. The overall critical path is T_{MUX} as drawn in dash line. Moreover, the multiplexers A_{02} , A_{12} , A_{22} , A_{32} , B_{02} , B_{12} , B_{22} , and B_{32} suffer from the large fanout. In order to achieve the desired throughput rate, the architecture of Fig. 8 is retimed as shown in Fig. 9. We discuss some design efforts in detail as follows:

- 1) *Employ inverted multiplexer*: The delay of a multiplexer with an inverted output is faster than a multiplexer without inverter output in the standard cell library. For the clarification of presentation, we call a multiplexer with an inverted output an inverted multiplexer. In UMC 0.18- μm cell libraries, the latency of an inverted multiplexer can be a half of a multiplexer and is denoted T_{muxi} . Moreover, when the selector of a multiplexer is inverted, we just exchange the two inputs. Then, the output of the multiplexer will be the same. Therefore, we replace some multiplexers with inverted multiplexers in our design. While the outputs of the multiplexers A_{02} , A_{12} , A_{22} , and A_{32} is added by the sum of first stage pre-computation, we do not replace these multiplexers with inverted multiplexers; otherwise, the output values will be changed. On the other hand, the fanout number of these multiplexers is 16; therefore, a multiplexer with large driving capacity is chosen.

Fig. 8. Architecture of the proposed 4-unfolding FBF.

- 2) *Isolate large fanout by inserting buffers*: As showing in Fig. 8, the fanout of the multiplexers B_{02} , B_{12} , B_{22} , and B_{32} is also very large. To shorten the latency, we apply one inverter and buffer to increase the driving capacity, as showing in Fig. 9. By doing so, we can mitigate the degradation of the delay time in the critical path due to the large fanout. For example, the fanout of multiplexer B_{02} is multiplexers A_{02} , A_{11} , A_{20} , B_{30} , B_{21} , and B_{12} . We insert one buffer to drive A_{11} and cascade another to A_{20} , which has the largest loading capacity. This can reduce the loading of multiplexer. Then, the critical path through B_{02} , A_{02} , VM, B_{00} , and B_{01} is reduced. Also, we apply one inverter to drive the buffer and the rest multiplexers.
- 3) *Retiming scheme*: Although we can reduce the latency from B_{02} to A_{02} (example above), the latency from B_{02} to A_{11} and A_{20} increase. To solve the problem, we can apply the retiming approach. For example, the latency from B_{02} to A_{11} is two multiplexers, one inverter, and one buffer delay time. Therefore, the delay element D_1 must be moved to the front of the VM Unit. Then, the latency from B_{10} to A_{12} is close to from B_{02} to A_{12} . Therefore, the effect of the delay time of the inserted buffer can be eliminated. D_2 and D_3 are retimed by using the same way. The architecture of Fig. 9 is a retimed version of Fig. 8.

In summary, after some modifications, the critical path is in dash line and the latency is 0.9 ns, which is evaluated by using

Fig. 9. Proposed 4-unfolding FBF after applying the retiming approach and some modifications.

TABLE II

USED CELLS IN THIS PAPER ARE LISTED AND THE DESCRIPTIONS OF CELL TYPES CAN BE LOOKED UP IN UMC 0.18- μ m CELL LIBRARY DATABOOK

	[0][0]	N -Partial pre-computation	Two-Stage pre-computation
Number of Adders	$(M)^L$	$2^{N+M} + (L-N+1)$	$2(M)^{(L/2)}$
Number of 2-to-1 Multiplexers	$(M)^{L-1}$	$(2^{N+M}-1)$	$2(M)^{L/2-2}$
Number of Multiplier	0	$(L-N+1)$ $(\log_2 M)$ -bits	0
Critical Path	$(\log_2 M)T_{mux}$	$T_{FA} \sim T_{Multiplier}$	$1/(L/2+1)T_{add} + (\log_2 M)T_{mux}$

UMC 0.18- μ m cell libraries. It can provide the delay element with margin enough. The extra overhead is only 64 adders and 56 2-to-1 multiplexers. The types of the used cells are listed in Table II.

V. CONCLUSION

In this brief, we proposed two new schemes to speed up the computation of a FBF for high performance DFE designs. The proposed architecture can reduce the hardware overhead extremely, and the latency is increased slightly. The partial pre-computation can trade-off between hardware complexity and

computation speed. Two-stage pre-computation can be used for very high speed applications and applied to 10 Gbase-LX4 optical communication systems.

REFERENCES

- [1] K. K. Parhi, "Pipelining in algorithms with quantizer loops," *IEEE Trans. Circuits Syst.*, vol. 37, no. 7, pp. 745–754, Jul. 1991.
- [2] S. Kasturia and J. H. Winters, "Techniques for high-speed implementation of nonlinear cancellation," *IEEE J. Sel. Areas Commun.*, vol. 9, pp. 711–717, Jun. 1991.
- [3] P. M. Kogge, *The Architecture of Pipelined Computers*. New York: McGraw-Hill, 1981.
- [4] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filter- Part I and II," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 1099–1135, Jul. 1989.
- [5] J. I. Acha, "Computational structures for fast implementation of L-path and L-block digital filters," *IEEE Trans. Circuits Syst.*, vol. 36, no. 6, pp. 805–812, Jun. 1989.
- [6] L. E. Lucke and K. K. Parhi, "Parallel processing for rank order and stack filter," *IEEE Trans. Signal Process.*, no. 5, pp. 1178–1189, May 1994.
- [7] C. Leiserson, F. rose, and J. Saxe, "Optimizing synchronous circuitry by retiming," in *Proc. 3rd Caltech Conf. VLSI*, 1983, pp. 87–116.
- [8] 10GBASE-LX4, IEEE Std 802.3ae-2002, <http://www.ieee802.org/3/ae>.
- [9] K. K. Parhi, "Fast low-power VLSI binary addition," in *Proc. IEEE Int. Conf. Comput. Design (ICCD'97)*, Oct. 1997, pp. 676–684.
- [10] —, "Pipelining of parallel multiplexer loops and decision feedback equalizers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP '04)*, May 2004, vol. 5, pp. 17–21.