Techniques for High-Speed Implementation of Nonlinear Cancellation

Sanjay Kasturia, *Member, IEEE*, and Jack H. Winters, *Senior Member, IEEE*

Abstract—**Nonlinear cancellation (NLC) can significantly reduce intersymbol interference (ISI) in long-haul direct-detection fiber-optic systems, and can thereby more than double the dispersion-limited data rate and/or distance. The ISI reduction by NLC is achieved by subtracting the interference caused by previously detected symbols; this subtraction, however, requires previous decisions to be fed back to the decision element, and delays in the feedback loop can severely limit the maximum data rate of a detector operating with NLC. In this paper, we present techniques for breaking the bottleneck caused by the feedback loop. First, we show how to simplify the loop to avoid high-speed switching of analog signals by using multiple decision elements, each with a different threshold level. We then show how to use lookahead computation to increase the delay permissible in the feedback loop. These techniques permit NLC to be implemented in integrated-circuit form at rates limited only by the detector switching speeds. These techniques are also useful in other applications requiring speedup of feedback loops with decision elements in the loop (e.g., decision feedback equalizers).**

I. Introduction

NONLINEAR cancellation (NLC) can greatly reduce pattern-dependent intersymbol interference (ISI) in a receiver by subtracting interference caused by previously detected symbols. With the rapid development of optical amplifiers, ISI becomes a major factor limiting data rates in long-haul direct-detection fiber-optic systems; NLC offers the potential of increasing the maximum data rate of these systems. This potential can only be realized if we devise techniques for implementing NLC, preferably in integrated circuits, at the high rates desired (gigabits/s).

Previous papers [1], [2] have studied the reduction of ISI achievable with NLC. It was shown in [2] that NLC can more than double the maximum data rate of gigabits per second (Gb/s) lightwave systems by canceling post-cursor ISI[1]. Because NLC is a nonlinear technique, it can handle ISI that arises from a variety of impairments, such as transmit laser chirp, chromatic dispersion, polarization dispersion, and nonideal receiver frequency response. NLC can also reduce ISI caused by bandwidth limitations of the detector itself (when it is operated at high data rates), thereby increasing the maximum *useful* operating rate of the detector. However, NLC, as described in [1], [2], is difficult to implement at high data rates. Fig. 1 shows a simple nonlinear canceler. We see that the canceler in Fig. 1 requires an analog signal to be switched at the data rate in the feedback loop. Such a system is difficult to implement because switching transients tend to corrupt the analog signal. In addition, propagation delays of elements in the feedback loop limit the maximum data rate of the detector with NLC. Even though the comparator may be able to operate faster, the data rate is limited to the inverse of the maximum propagation and processing delay around the feedback loop because the signal being fed back must arrive at the comparator before the next symbol. This speed limitation, imposed by the presence of feedback, can be expressed as an iteration bound[2] (see Parhi and Messerschmitt [5]). For NLC to be useful in high data rate systems, these problems must be overcome.

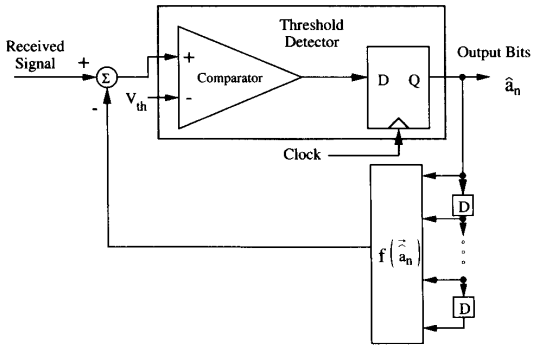


Fig. 1. A simple nonlinear canceler.

In this paper, we develop implementation techniques for NLC that eliminate these problems at the expense of increased circuit complexity. Integrated circuits for decision-feedback based equalization at rates of several gigabits/s will be easily implemented using the techniques developed in this paper. Multiple detectors (2^N,

Manuscript received May 30, 1990; revised March 14, 1991.

The authors are with the Network Systems Research Department, AT&T Bell Laboratories, Holmdel, NJ 07733.

IEEE Log Number 9100128.

[1]ISI caused by symbols preceding the one currently at the decision element. NLC can be effective in reducing ISI even in systems with precursor ISI because the relative strengths of precursor and post-cursor ISI can be altered by (a) changing the decision time, and/or (b) using a tapped-delay line linear equalizer [3], [4].

[2]The iteration bound on the throughput of a system with feedback loops is given by the minimum, over all loops, of D_i/C_i where D_i is the total number of logical delays in the ith loop, and C_i is the total latency required for all the computations in the same loop [5].

where N is the number of bits fed back), each with a different threshold level, are used to avoid the switching of analog signals. The selection of the threshold level (i.e., the feedback signal) occurs after threshold detection. This is accomplished by choosing the output of the detector corresponding to the correct threshold rather than by switching analog signals. The selection algorithm is then iterated L times so that the maximum delay permissible in generating the feedback signal is increased from one symbol period to L symbol periods. We then pull some of the computation outside the loop (using *lookahead* computation) to minimize propagation delay in the feedback loop. With these techniques, NLC can be easily implemented *on the detector chip* to increase the maximum data rate of Gb/s communication systems and/or detectors.

In Section II, we briefly describe nonlinear cancellation, and outline some of the problems of implementing NLC at high data rates. Techniques for overcoming these problems are developed in Section III and are illustrated with the example of a simple one-tap NLC. Section IV briefly describes how to generalize the techniques to more complex NLC's and presents some more examples. Section V summarizes this paper.

II. The Nonlinear Canceler

Fig. 1 shows an implementation of NLC as discussed in [2]. The threshold detector compares the received signal to the threshold (V_{th}) to determine the output bit $\hat{a}_n$. Before making the comparison, the NLC generates a correction signal $f(\vec{a})$ based on N previously detected bits[3]

$$\vec{\hat{a}}_n = (\hat{a}_{n-1}\hat{a}_{n-2} \cdots \hat{a}_{n-N}).$$

This signal is fed back to the input to compensate for the ISI caused by bits transmitted prior to the bit currently at the detector. We will make the assumption that the previous bits have been detected correctly, and will, for convenience, use a_n instead of the more correct $\hat{a}_n$ to denote the detected bits. The reader should note that the assumption of correct prior decisions ensures that NLC will reduce ISI. Whereas this assumption is not entirely correct, NLC does significantly improve performance in spite of occasional (though rare) errors in the feedback path. This issue of errors in the feedback path has been addressed by others [7], [8], and we will not dwell on it any further.

Before proceeding further, we will modify Fig. 1 by moving the feedback signal to the threshold input of the comparator as shown in Fig. 2. This is done to avoid interaction between the received and feedback signals. Here we see that the feedback signal can take one of many values, none of which is related to the voltages used to represent the logic values. At low data rates, such a signal can be generated by a random access memory followed by a digital-to-analog converter (see [9]). However, this technique is impractical at rates of Gb/s. Swartz has devised a circuit [10] for a variable gain latch to be used with the NLC (at Gb/s rates) as shown in Fig. 3 (for the case of $N = 1$). The output of the variable gain latch $\overline{Q}$ switches between V_0 and V_1, which are the appropriate threshold values corresponding to a previously detected *zero* or *one*. $\overline{Q}$ must have a large dynamic range, should not have any large transients, and must have low noise levels. These are severe constraints, and practical circuits may be unable to satisfy all of them at Gb/s data rates. This approach also becomes more difficult to implement as more bits are used in the feedback path (larger N) because then the latch must be able to switch between 2^N values.

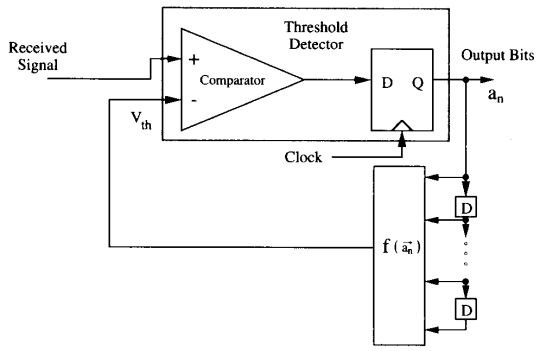


Fig. 2. Detector with feedback moved to threshold.

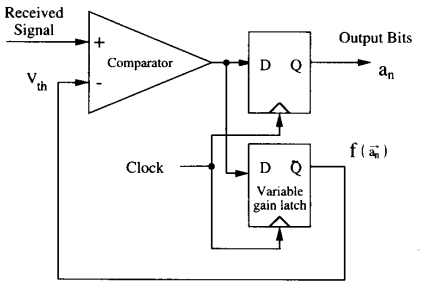


Fig. 3. NLC with variable gain latch.

Apart from the problem of generating the analog feedback signal, NLC, as shown in Figs. 1, 2, and 3, has its speed limited to the reciprocal of the total propagation and processing delay around the feedback loop. Basically, the decision on the nth bit has to propagate around the feedback loop and appear at the input to the comparator before the comparator begins processing the next symbol. Fig. 4 depicts the timing requirement pictorially. The speed limitation imposed by the feedback loop in NLC is severe because the propagation delay of just the threshold detector itself can be greater than one symbol period in a high-speed detector [11], [12]. The timing constraints imposed by the presence of feedback will also reduce the clock-jitter tolerance of the system.

[3] If $f(\cdot)$ is a linear function of $\vec{a}$, (i.e., $f(\vec{a}_n) = \Sigma_{i=1}^{N} w_i \hat{a}_i$), the technique is referred to as decision feedback equalization (DFE) [6]. Because DFE is a subset of NLC, the techniques described in this paper are also useful with DFE.

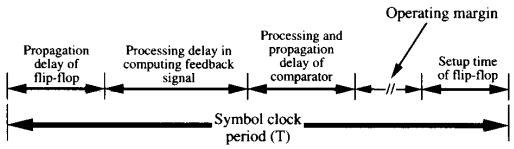


Fig. 4. Timing in the feedback loop in NLC.

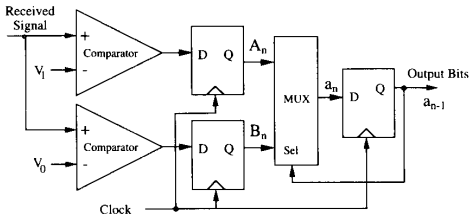


Fig. 5. Removing analog feedback.

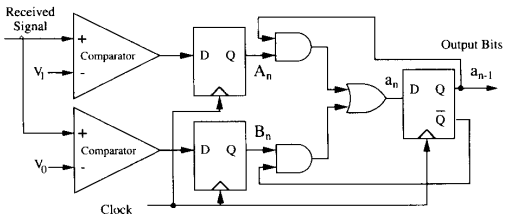


Fig. 6. Removing analog feedback: gate level.

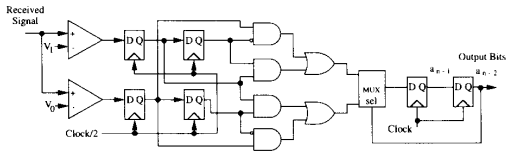


Fig. 7. Increasing permissible delay in the feedback loop.

III. Implementation Techniques

A. Avoiding Analog Switching

A look at Fig. 2 reveals that at most 2^N distinct threshold values may be fed back to the comparator. One way of avoiding this threshold feedback is to use 2^N distinct comparators, with each using one of the 2^N possible threshold values. Now the problem of feeding back the correct threshold (which depends on the previously detected bits) is replaced by selecting the comparator which uses the correct threshold value. Fig. 5 shows a block diagram of such a system; note that we have assumed only one feedback tap ($N = 1$), and hence we need two comparators. Fig. 5 shows a 2:1 multiplexer being used to select between the two comparators. V_0 and V_1 are the two threshold values corresponding to the previous bit being a *zero* or a *one*. Fig. 6 incorporates a gate-level implementation of the 2:1 multiplexer. An integrated circuit implementation may combine the gates of the multiplexer with the gates implementing the latch in the *D* flip–flop.

Although Figs. 5 and 6 avoid analog feedback, they have the same time available (T) for processing and propagation around the feedback loop as the implementation in Fig. 3. However, the threshold detector and variable gain latch are no longer in the feedback loop. Instead, we have a multiplexer; because the propagation and processing delay is usually less for a multiplexer than for the components it has replaced, the system shown in Fig. 5 should be able to operate faster than the system in Fig. 3. The next subsection shows how to increase the time permitted for processing and propagation in the feedback loop.

B. Eliminating the Feedback Imposed Bottleneck

Going back to Fig. 5, we see that the nth bit is given by

$$a_n = A_n a_{n-1} + B_n \bar{a}_{n-1} \tag{1}$$

where A_n and B_n are outputs of the two comparators at the nth symbol period, and the overbar denotes a logical complement. Note that all the variables are Boolean, with values *zero* or *one*. If we write out the corresponding equation for a_{n-1}, we have

$$a_{n-1} = A_{n-1} a_{n-2} + B_{n-1} \bar{a}_{n-2}. \tag{2}$$

Using (2) to substitute for a_{n-1} in (1), we get

$$\begin{aligned} a_n &= A_n(A_{n-1}a_{n-2} + B_{n-1}\bar{a}_{n-2}) \\ &\quad + B_n\overline{(A_{n-1}a_{n-2} + B_{n-1}\bar{a}_{n-2})} \\ &= (A_nA_{n-1} + B_n\bar{A}_{n-1})a_{n-2} \\ &\quad + (A_nB_{n-1} + B_n\bar{B}_{n-1})\bar{a}_{n-2} \qquad (3) \\ &= f_1(n)a_{n-2} + f_2(n)\bar{a}_{n-2} \qquad (4) \end{aligned}$$

where

$$f_1(n) = A_nA_{n-1} + B_n\bar{A}_{n-1}$$

$$f_2(n) = A_nB_{n-1} + B_n\bar{B}_{n-1}.$$

Equation (3) represents a_n in terms of the A's, B's, and a_{n-2}, whereas (1) represents a_n in terms of a_{n-1}. The major significance of this difference is that an implementation based on (3) will permit a total propagation and processing delay of $2T$ in the feedback loop, whereas an implementation based on (1) would permit only T. Equation (3) looks a lot more complicated than (1), and one might assume that although it permits a larger delay, the increased complexity will take up the increase in permitted delay, resulting in a circuit which may be as slow as one implementing (1). However, this is not the case because $f_1(n)$, and $f_2(n)$ can be computed outside the feedback loop (*lookahead* computation) and then be provided as inputs to the feedback loop. The feedback loop itself

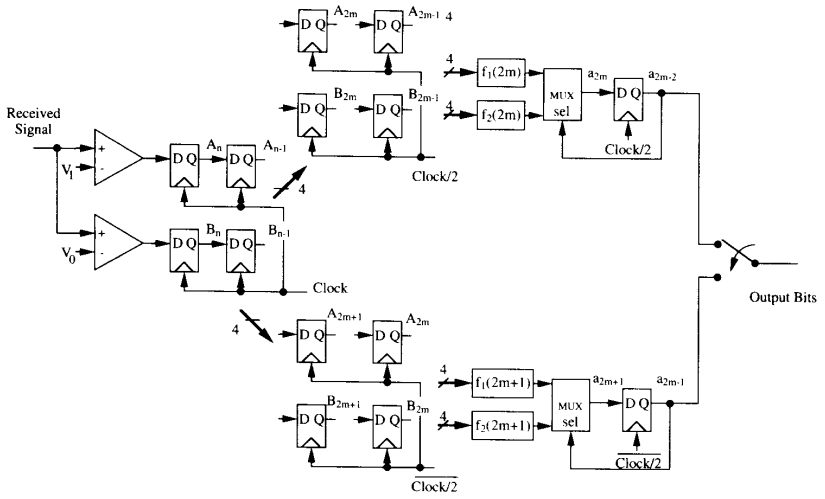


Fig. 8. Multiple feedback loops operating at lower speed.

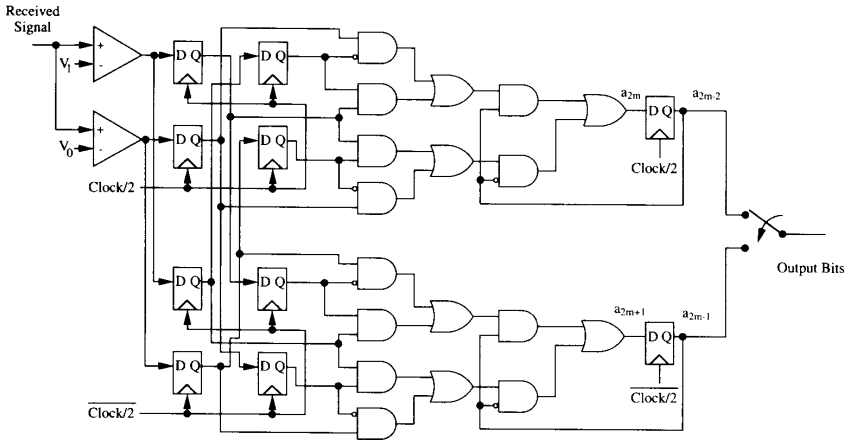


Fig. 9. A more detailed diagram.

must just implement (4) [see Fig. 7], which is quite simple and is identical to the feedback loop in Fig. 6.

The extra delay available in the loop in Fig. 7 may be used in two ways. First, we can use it to add a latch within the multiplexer to pipeline the multiplexer and increase its speed of operation. Second, it can be used to reduce the clock rate within the loop to half the data rate. In the later approach, the reader must note that (3) represents a_n in terms of a_{n-2}, and therefore if we use this equation repeatedly, starting with a_0, we will compute $a_2, a_4, \cdots, a_{2m}$. To compute the missing terms, which are the a_{2m+1}'s, we must iterate the same equation, but this time we must do it starting with a_1. Hence, we must replicate the feedback loop twice. Such a system, implementing (4), is illustrated in Fig. 8. Fig. 9 shows the same system at the gate level with a few simple changes to aid implementation. In an integrated circuit implementation, the gates for the multiplexer may be partly incorporated in the gates used to implement the D flip–flop which follows the multiplexer.

We have just illustrated a technique to increase the permissible processing and propagation delay in the feedback loop from T to $2T$ without increasing the circuit complexity of the feedback loop. We did this by "iterating" (1) twice. Similarly, by iterating (1) L times, we can increase the permissible delay to LT, and have L loops operating in parallel. Parhi, Meng, and Messerschmitt [5], [13] have used similar techniques, i.e., iterating a recursive relationship, to avoid feedback-imposed speed limitations for recursive and adaptive filters. However, their techniques are limited to instances of linear recursion. The vectorization of the *joint process estimator* carried out in [13]

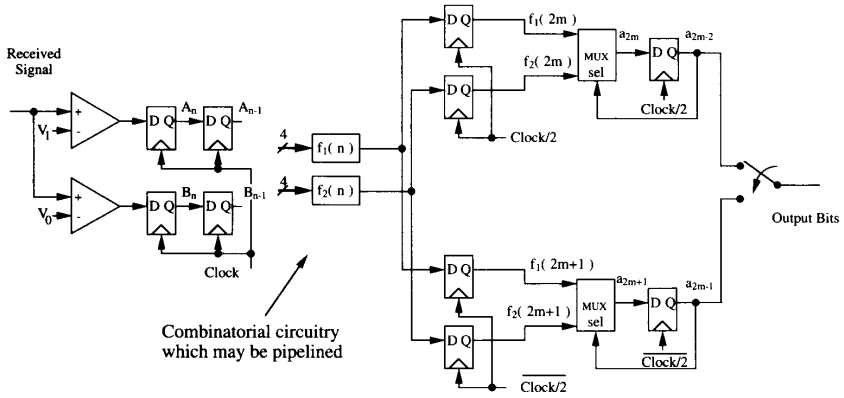


Fig. 10. Sharing the precomputation circuitry.

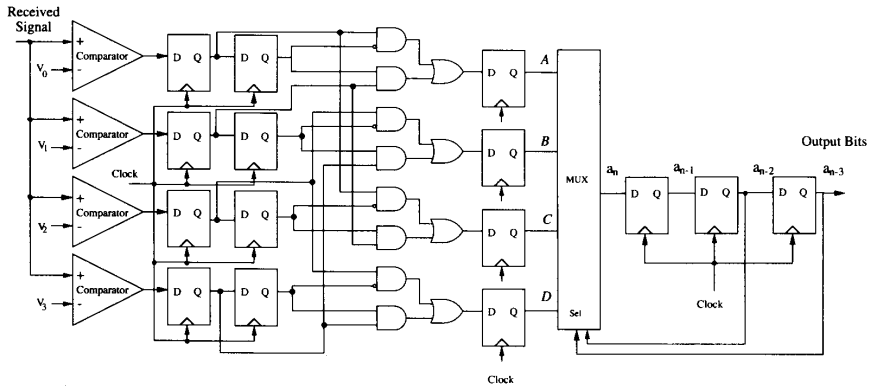


Fig. 11. Two-bit feedback: iterated twice.

breaks down with the introduction of decision feedback. Our approach to the problem is capable of handling recursion with decisions in the feedback loop.

1) Sharing Some Circuitry: In Figs. 8 and 9, the circuitry to precompute f_1 and f_2 has also been replicated. If this combinatorial circuitry is fast enough to operate at the symbol rate T, we can share this circuitry between the feedback loops instead of replicating it. This concept is illustrated in Fig. 10. If necessary, we can pipeline [14] this combinatorial circuitry to speed it up. Pipelining (by adding flip–flops between serial processing elements) will add to the propagation delay and will increase the overall latency through the receiver but it will not result in a data rate limitation because this circuitry is not in the feedback path.

IV. Generalizations

In the preceding section, we illustrated an L-fold increase (with examples for $L = 2$) of the permissible delay in the feedback loop for a NLC with one-tap feedback. For the more general case, with N feedback taps, 2^N distinct threshold values are possible. To avoid analog feedback, we must replicate the comparator 2^N times. If $A_{(i,n)}$ denotes the output of the ith comparator during the nth symbol period, a_n is given by

$$\begin{aligned} a_n &= A_{(1,n)}a_{n-1}a_{n-2}\cdots a_{n-N} \\ &\quad + A_{(2,n)}a_{n-1}a_{n-2}\cdots \bar{a}_{n-N} \\ &\quad + \cdots + A_{(2^N,n)}\bar{a}_{n-1}\bar{a}_{n-2}\cdots \bar{a}_{n-N}. \end{aligned} \tag{5}$$

To increase the permissible delay in the feedback loop to LT, we must iterate (5) L times. Each iteration involves selecting the a with the highest subscript from the right side of the equation, and then substituting for it in terms of previous bits using the relationship of (5). After iterating L times, we can expand the right-hand side into a sum-of-products form and group the terms as shown below:

$$\begin{aligned} a_n &= f_1(n)a_{n-L}a_{n-L-1}\cdots a_{n-L-N+1} \\ &\quad + f_2(n)a_{n-L}a_{n-L-1}\cdots \bar{a}_{n-L-N+1} + \cdots \\ &\quad + f_{2^N}(n)\bar{a}_{n-L}\bar{a}_{n-L-1}\cdots \bar{a}_{n-L-N+1} \end{aligned} \tag{6}$$

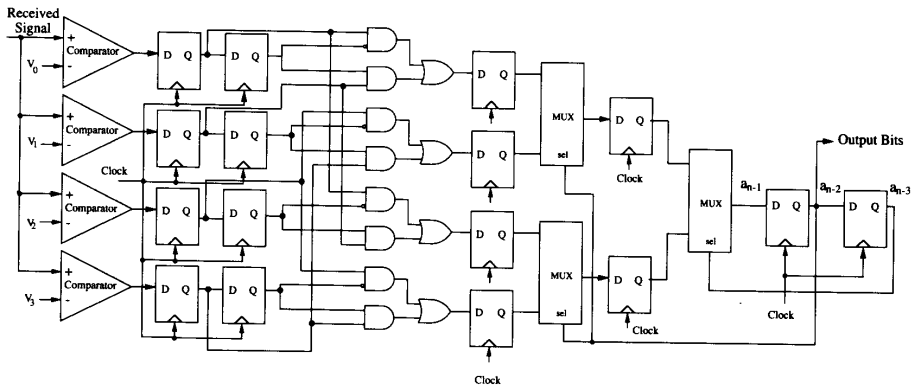


Fig. 12. Two-bit feedback: pipelined loop.

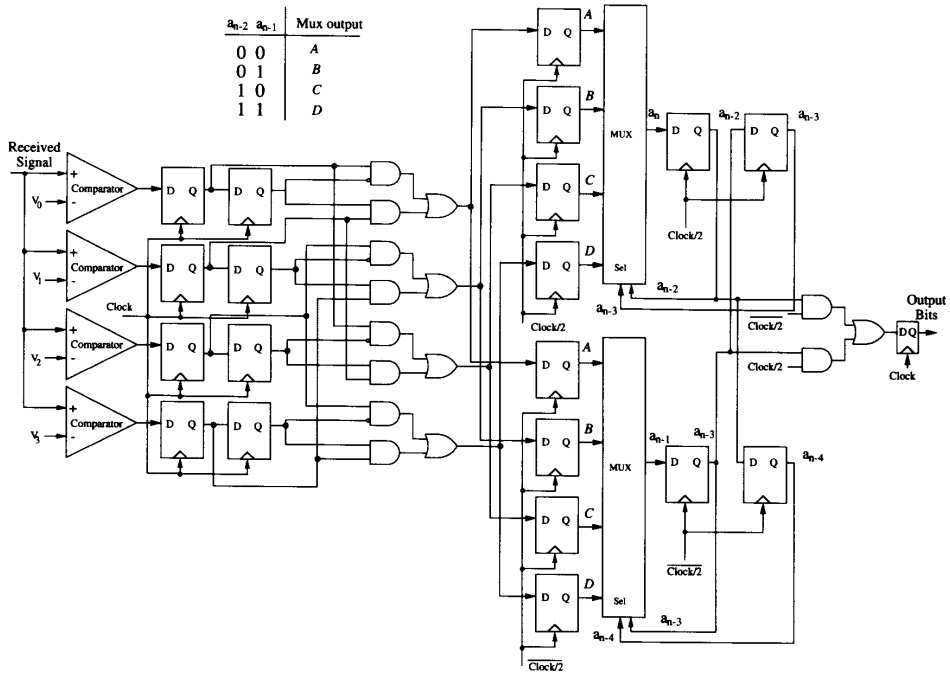


Fig. 13. Two-bit feedback: parallel loops.

where each $f_i(n)$ is a function of $A_{(j,n)}$, $A_{(j,n-1)} \cdots$, $A_{(j,n+1-L-N)}$, $j = 1, \cdots 2^N$. The $f_i(n)$ can be precomputed outside the feedback loop. Basically, the nth output bit is given by one of the f_i with the selection based on N previous bits delayed by L symbol periods. Note that to write the iterated (5) in the form shown in (6), we first expand the right-hand side of the iterated equation into a sum-of-products form. A product term which contains more than one occurrence of an a_i or its inverse is then simplified to contain only one occurrence. This is always possible because the a_i are *zero* or *one*. $a_i a_i$ is equal to a_i, and $\bar{a}_i \bar{a}_i$ equals $\bar{a}_i$, and any product terms which have $a_i \bar{a}_i$ are dropped because this is always *zero*.

We could alternatively rewrite (6) as the sum of a selection of terms from its right-hand side:

$$a_n = f_i(n) \tag{7}$$

where $(i - 1)$ is given by the N-bit binary representation [rather than the logical expression as in (6)]

$$a_{n-L} a_{n-L-1} \cdots a_{n-L-N+1}$$

or

$$i = 1 + a_{n-L} a_{n-L-1} \cdots a_{n-L-N+1}. \tag{8}$$

As in Section III, the L-fold increase in the permissible loop propagation and processing delay can be used either

to pipeline the multiplexer, or to reduce the clock rate in the feedback loop. With the second approach, the feedback loop itself must be replicated L times, with each loop running at $1/L$th the symbol rate. Because the loops run at $1/L$ times the symbol rate and are no more complex than the original loop, our technique can achieve an L-fold increase in the speed of operation of a receiver with NLC. The precomputation of the $f_i(n)$'s can be pipelined so the combinatorial circuitry is shared between the L feedback loops. Figs. 11–13 sketch the applications of our techniques to a system with two-bit feedback. Although we have assumed binary symbols in all of the previous sections, as a further generalization, our techniques can be easily extended to M-ary signaling.

V. Summary and Conclusion

In summary, by using multiple detectors with digital logic, we can eliminate the analog switch and propagation delay problems of nonlinear cancellation and decision feedback equalization. 2^N detectors are required with N feedback bits, but the additional circuitry required for higher data rates is all-digital logic and thus easily amenable to integration. Thus, we can achieve higher speed at the expense of increased circuit complexity. With these techniques, NLC and DFE can be easily implemented on the detector chip to increase the maximum data rate of Gb/s communication systems and/or detectors.

References

[1] E. Biglieri, A. Gersho, R. D. Gitlin, and T. L. Lim, "Adaptive cancellation of nonlinear intersymbol interference for voiceband data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 765–777, Sept. 1984.

[2] J. H. Winters and R. D. Gitlin, "Electrical signal processing techniques in long-haul fiber-optic systems," *IEEE Trans. Commun.*, vol. 38, pp. 1439–1453, Sept. 1990.

[3] J. H. Winters and M. A. Santoro, "Experimental equalization of polarization dispersion," *IEEE Photon. Technol. Lett.*, vol. 2, no. 8, pp. 591–593, Aug. 1990.

[4] J. H. Winters, "Equalization in coherent lightwave systems using a fractionally spaced equalizer," *J. Lightwave Technol.*, vol. 8, no. 10, pp. 1487–1491, Oct. 1990.

[5] K. K. Parhi and D. G. Messerschmitt, "A bit parallel bit level recursive filter architecture," in *Proc. Conf. Comput. Design*, Oct. 1986, pp. 284–289.

[6] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1983.

[7] D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "An upper bound on the error probability in decision feedback equalizers," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 490–497, July 1974.

[8] R. A. Kennedy, B. D. O. Anderson, and R. R. Bitmead, "Tight bounds on the error probabilities of decision feedback equalizers," *IEEE Trans. Commun.*, vol. COM-35, pp. 1022–1029, Oct. 1987.

[9] K. D. Fisher, J. M. Cioffi, and C. M. Melas, "An adaptive decision feedback equalizer for storage channels suffering from nonlinear ISI," in *Proc. Int. Conf. Commun.*, Boston, MA, June 1989, pp. 53.7.1–53.7.5.

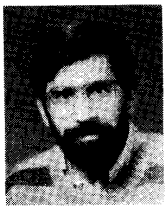
[10] R. D. Gitlin, R. G. Swartz, and J. H. Winters, "Nonlinear cancellation techniques," private communication.

[11] AT&T LG1088AX Decision Circuit: 1.7 Gb/s, Preliminary Data Sheet.

[12] GigaBit Logic 10G021A, Data Sheet.

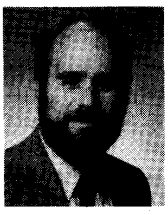
[13] T. Meng and D. G. Messerschmitt, "Arbitrarily high sampling rate adaptive filters," *IEEE Trans. Signal Process.*, vol. 35, pp. 455–470, Apr. 1987.

[14] J. R. Jump and S. R. Ahuja, "Effective pipelining of digital systems," *IEEE Trans. Comput.*, vol. C-27, pp. 855–865, Sept. 1978.



Sanjay Kasturia (S'89–M'89) received the B.Tech. degree from the Indian Institute of Technology, Kanpur, India, in 1981, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, where he was partly supported by an IBM graduate fellowship.

He has worked at Magnum Microwave, Vitalink Communications, Cylink, and is currently a member of the Network Systems Research Department of AT&T Bell Laboratories, Holmdel, NJ.



Jack H. Winters (S'77–M'82–SM'88) was born in Canton, OH, on September 17, 1954. He received the B.S.E.E. degree (summa cum laude) from the University of Cincinnati, Cincinnati, OH, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from The Ohio State University, Columbus, in 1978 and 1981, respectively.

From 1973 to 1976, he was with the Communications Satellite Corporation, Washington, DC, and from 1977 to 1981 with the ElectroScience Laboratory, The Ohio State University, where he studied adaptive antenna arrays. Since 1981, he has been with AT&T Bell Laboratories, Holmdel, NJ, where he is in the Network Systems Research Department. He has been involved in research on modulation and coding, mobile and indoor radio systems, neural networks, TV signal processing, and applications of high-T_c superconductors in communication systems. Currently, he is involved in research on signal processing for lightwave systems.

Dr. Winters is a member of Sigma Xi. He received The Ohio State University ElectroScience Laboratory's award for outstanding dissertation of 1981.