

Recursive All-Lag Reference-Code Correlators

Tung-Sang Ng, Kun-Wah Yip, and Chin-Long Cheng

Abstract—An all-lag reference-code correlator generates an all-lag even- or odd-correlation vector at a rate equal to the rate of incoming data samples. Direct implementation of an all-lag reference-code correlator requires N parallel correlators, and the resultant degree of complexity is of the order N^2 , where N is the length of the reference code. This paper derives two recursive forms for all-lag reference-code correlators. One generates all-lag even correlation and the other one generates all-lag odd correlation. It is shown that the proposed recursive all-lag reference-code correlator can be implemented with a complexity approximately equal to that of a single parallel correlator. That is, the degree of complexity of the proposed recursive all-lag reference-code correlator is of the order N . Thus, substantial reduction in the implementation complexity is achieved.

Index Terms—All-lag reference-code correlator, bank of serial correlators, low-complexity implementation, parallel correlator, recursive relationship, serial correlator, spread spectrum.

I. INTRODUCTION

Correlators are widely used in applications involving signals that are formed by periodic repetition of reference codes with or without data modulation. The signals may be further corrupted by noise and various kinds of interference. Depending on applications, the reference code can be a pseudonoise sequence, a sampled sinusoidal wave or, in fact, any arbitrary sequence of data. As a particular example, the reference code in a direct-sequence spread-spectrum (DSSS) system is a pseudonoise spreading sequence. DSSS techniques [1], [2] have applications in many areas such as multiple-access data communications, secure communications, channel sounding, ranging and target identification using radars or sonars, and navigation using global positioning system (GPS). A correlator is required in a DSSS receiver to initially acquire the incoming DSSS signal. It is also used to perform other functions such as code tracking, symbol and carrier clock recovery, demodulation of information symbols embedded in a DSSS signal, and channel estimation.

In this paper, we are concerned with all-lag reference-code correlators. An all-lag reference-code correlator correlates a stream of data samples $\{d_n\}$ with $0, 1, \dots, N-1$ lags of a length- N reference code sequence $\{c_0, c_1, \dots, c_{N-1}\}$ and thereby produces a stream of all-lag even-correlation vectors $\{\mathbf{r}_n\}$ or a stream of all-lag odd-correlation vectors $\{\bar{\mathbf{r}}_n\}$ at a rate equal to the rate of incoming data samples. In this context, $\mathbf{r}_n = [r_{0,n}, r_{1,n}, \dots, r_{N-1,n}]^T$ and $\bar{\mathbf{r}}_n = [\bar{r}_{0,n}, \bar{r}_{1,n}, \dots, \bar{r}_{N-1,n}]^T$ are given by

$$\mathbf{r}_n = \mathbf{C} \mathbf{d}_n \quad (1)$$

and

$$\bar{\mathbf{r}}_n = \bar{\mathbf{C}} \mathbf{d}_n \quad (2)$$

Manuscript received August 1999; revised July 2000. This work was supported in part by the Hong Kong Research Grants Council and in part by the University Research Committee of the University of Hong Kong. This paper was recommended by Associate Editor J. Le Blanc.

The authors are with the Department of Electrical and Electronic Engineering, the University of Hong Kong, Hong Kong (e-mail: tsng@eee.hku.hk; kwyip@eee.hku.hk; clcheng@eee.hku.hk).

Publisher Item Identifier S 1057-7130(00)11034-1.

respectively, where

$$\mathbf{C} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-2} & c_{N-1} \\ c_{N-1} & c_0 & c_1 & \cdots & c_{N-3} & c_{N-2} \\ c_{N-2} & c_{N-1} & c_0 & \cdots & c_{N-4} & c_{N-3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ c_2 & c_3 & c_4 & \cdots & c_0 & c_1 \\ c_1 & c_2 & c_3 & \cdots & c_{N-1} & c_0 \end{bmatrix} \quad (3)$$

and

$$\bar{\mathbf{C}} = \begin{bmatrix} c_0 & c_1 & c_2 & \cdots & c_{N-2} & c_{N-1} \\ -c_{N-1} & c_0 & c_1 & \cdots & c_{N-3} & c_{N-2} \\ -c_{N-2} & -c_{N-1} & c_0 & \cdots & c_{N-4} & c_{N-3} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ -c_2 & -c_3 & -c_4 & \cdots & c_0 & c_1 \\ -c_1 & -c_2 & -c_3 & \cdots & -c_{N-1} & c_0 \end{bmatrix} \quad (4)$$

are $N \times N$ matrices, and

$$\mathbf{d}_n = [d_{n-(N-1)}, d_{n-(N-2)}, \dots, d_{n-1}, d_n]^T \quad (5)$$

is a data vector containing N most-recent data samples. Note that the subindex m of $r_{m,n}$ and $\bar{r}_{m,n}$ refers to a lag of the reference code sequence while the second subindex n is time. For a description of even- and odd-correlation functions, interested readers may refer to [3] and [4], and the references therein.

In the following, we shall show that all-lag reference-code correlators can be used to generate outputs of serial correlators, parallel correlators, and banks of serial correlators, the latter three types of correlators being commonly used in practical situations [5]–[13]. A serial correlator produces one correlation output every N data samples. Its outputs can be obtained from an all-lag reference-code correlator by $r_{0,n}$ s, n being a multiple of N . A parallel correlator provides more correlation information. It correlates the N most-recent data samples with the reference code and yields one correlation result at each sampling instant. In this regard, the correlation outputs are produced at the same rate as the incoming data samples. The outputs of a parallel correlator are therefore obtained by $r_{0,n}$ s where n is an integer. A bank of serial correlators consists of N serial correlators and is used for correlating a sequence of N data samples, wherein the other sequence used for correlation in the n th serial correlator $n = 0, 1, \dots, N-1$ is the reference code cyclic-shifted with n shifts. Hence, the outputs of a bank of serial correlators can be obtained by \mathbf{r}_n s where n is a multiple of N . Interest in all-lag reference-code correlators arises because they provide more correlation information than the above three types of correlators. The additional correlation information provided by an all-lag reference-code correlator, when processed, can be utilized for various purposes, for example, faster acquisition and more robust channel estimation. In Section II, we shall further elaborate the application advantages of all-lag reference-code correlators.

Direct implementation of an all-lag reference-code correlator is by means of N parallel correlators, where the m th parallel correlator $m = 0, 1, \dots, N-1$ correlates a block of data samples given by \mathbf{d}_n with the sequence taken from the m th row of \mathbf{C} or $\bar{\mathbf{C}}$ according to whether even- or odd-correlation values are to be generated, and produces a sequence of correlation results $\{r_{m,n}\}$ or $\{\bar{r}_{m,n}\}$ at a rate of one result per sampling instant. A practical method to implement a parallel correlator is based on the systolic array architecture [5], [6]. Fig. 1 depicts such a parallel correlator for generating, as an example $\{r_{0,n}\}$ and Table I lists the required numbers of multipliers, adders, etc., for implementing a

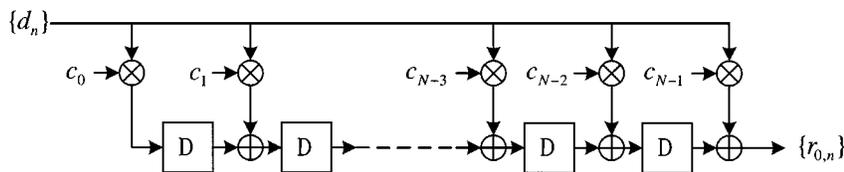


Fig. 1. Parallel correlator.

TABLE I
IMPLEMENTATION COMPLEXITY OF VARIOUS CORRELATORS

	multipliers	Required number of:		
		adders	storage units	negator
Parallel correlator	N	$N-1$	$N-1$	—
All-lag reference-code correlator directly realized by N parallel correlators	N^2	$N(N-1)$	$N(N-1)$	—
Recursive all-lag reference-code correlator that generates $\{r_n\}$	N	$N+1$	$2N$	1
Recursive all-lag reference-code correlator that generates $\{\bar{r}_n\}$	N	$N+1$	$2N$	1

parallel correlator. It is apparent that a parallel correlator comprises N multipliers, $N-1$ adders, and $N-1$ storage elements, so that the degree of implementation complexity is of the order N . Since an all-lag reference-code correlator implemented using the direct approach comprises N parallel correlators, the degree of implementation complexity is of the order N^2 . The resultant implementation complexity is especially significant when the reference-code length N is large.

Previous research effort has been devoted to minimize the implementation complexity of a parallel correlator for some special cases [14]–[17]. However, techniques directed to the reduction of implementation complexity for an all-lag reference-code correlator in general have not appeared in the previous literature. The objective of the present work is to develop a low-complexity architecture for an all-lag reference-code correlator. Based on (1) and (2), we derive two recursive forms, one for even and another one for odd correlations. All-lag reference-code correlators that are realized by these recursive relationships are referred to as recursive all-lag reference-code correlators. In this paper, we show that they can be efficiently realized with a complexity approximately equal to that of a single parallel correlator. That is, the resultant implementation complexity is of the order N . This result enables system designers to utilize all-lag correlation information while keeping the implementation cost low.

The rest of the paper is organized as follows. Recursive forms for all-lag reference-code correlators that generate $\{r_n\}$ and $\{\bar{r}_n\}$ are derived in Sections III and IV, respectively. Implementation aspects of recursive all-lag reference-code correlators are also discussed. Conclusions are drawn in Section V.

II. ADVANTAGES OF ALL-LAG CORRELATORS

We shall illustrate the advantages of all-lag correlators by considering the acquisition process of a DSSS signal [18]. In particular, we shall indicate the advantages of using all-lag correlators over using parallel correlators.

Consider first the case of using a parallel correlator which correlates a sequence of DSSS signal samples $\{d_n\}$ with a reference code $\{c_n\}$

and generates a sequence of correlation values $\{u_n\}$ at a rate equal to the rate of incoming signal samples, where

$$u_n = c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + c_2 d_{n-(N-3)} + \cdots + c_{N-2} d_{n-1} + c_{N-1} d_n \quad (6)$$

is the correlation result obtained at the n th sampling instant. The reference-code length N is normally selected such that it is equal to the length of the spreading sequence multiplied by the number of samples per chip. Before acquisition, the DSSS signal is not code-aligned with the receiver's copy of the reference code sequence. Since the reference-code length is N , we can code-align, or acquire, the incoming DSSS signal at the receiver by computing N correlation values corresponding to the correlation of the signal with $0, 1, \dots, N-1$ lags (or delays) of the reference code. The receiver is therefore required to compute

$$\begin{aligned} u_n &= c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + c_2 d_{n-(N-3)} \\ &\quad + \cdots + c_{N-2} d_{n-1} + c_{N-1} d_n \\ u_{n+1} &= c_0 d_{n-(N-2)} + c_1 d_{n-(N-3)} + c_2 d_{n-(N-4)} \\ &\quad + \cdots + c_{N-2} d_n + c_{N-1} d_{n+1} \\ u_{n+2} &= c_0 d_{n-(N-3)} + c_1 d_{n-(N-4)} + c_2 d_{n-(N-5)} \\ &\quad + \cdots + c_{N-2} d_{n+1} + c_{N-1} d_{n+2} \\ &\quad \vdots \\ u_{n+N-2} &= c_0 d_{n-1} + c_1 d_n + c_2 d_{n+1} \\ &\quad + \cdots + c_{N-2} d_{n+N-3} + c_{N-1} d_{n+N-2} \\ u_{n+N-1} &= c_0 d_n + c_1 d_{n+1} + c_2 d_{n+2} \\ &\quad + \cdots + c_{N-2} d_{n+N-2} + c_{N-1} d_{n+N-1} \end{aligned} \quad (7)$$

and the acquisition circuit determines which one of these values has the largest magnitude. Acquisition is declared on the time position where the largest magnitude occurs. Notice that $2N-1$ data samples are

involved so that the time required to complete the acquisition process is $2N - 1$ sampling periods.

In the absence of data modulation embedded in the DSSS signal, the signal is a periodic repetition of the reference code sequence. The intended information contained in signal samples $d_{n+1}, d_{n+2}, \dots, d_{n+N-1}$ is also contained in $d_{n-(N-1)}, d_{n-(N-2)}, \dots, d_{n-1}$, respectively, so that (7) can be expressed as

$$\begin{aligned}
 u_n &= c_0 d_{n-(N-1)} + c_1 d_{n-(N-2)} + c_2 d_{n-(N-3)} \\
 &\quad + \dots + c_{N-2} d_{n-1} + c_{N-1} d_n \\
 u_{n+1} &= c_0 d_{n-(N-2)} + c_1 d_{n-(N-3)} + c_2 d_{n-(N-4)} \\
 &\quad + \dots + c_{N-2} d_n + c_{N-1} d_{n-(N-1)} \\
 u_{n+2} &= c_0 d_{n-(N-3)} + c_1 d_{n-(N-4)} + c_2 d_{n-(N-5)} \\
 &\quad + \dots + c_{N-2} d_{n-(N-1)} + c_{N-1} d_{n-(N-2)} \\
 &\quad \vdots \\
 u_{n+N-2} &= c_0 d_{n-1} + c_1 d_n + c_2 d_{n+(N-1)} \\
 &\quad + \dots + c_{N-2} d_{n-3} + c_{N-1} d_{n-2} \\
 u_{n+N-1} &= c_0 d_n + c_1 d_{n-(N-1)} + c_2 d_{n-(N-2)} \\
 &\quad + \dots + c_{N-2} d_{n-2} + c_{N-1} d_{n-1}. \tag{8}
 \end{aligned}$$

Thus, computation of $u_n, u_{n+1}, \dots, u_{n+N-1}$ is equivalent to computing \mathbf{r}_n given by (1). Rapid acquisition of the incoming DSSS signal is achieved by locating the time position having the largest magnitude among $r_{m,n}$, $m = 0, 1, \dots, N - 1$. It is apparent that acquisition of a DSSS signal by using \mathbf{r}_n can be achieved in a duration of N consecutive sampling periods while acquisition using a parallel correlator involves a larger data block of $2N - 1$ samples. All-lag correlators thus enable faster acquisition of DSSS signals.

In the presence of antipodal data modulation, that is, when the symbols are either $+1$ or -1 , successive symbols may or may not have a transition in polarity. When successive data symbols contained in DSSS signal samples $d_{n-(N-1)}, d_{n-(N-2)}, \dots, d_{n+N-1}$ have the same sign, it is easy to show that computation of $u_n, u_{n+1}, \dots, u_{n+N-1}$ is equivalent to the computation of \mathbf{r}_n . Acquisition is declared at the time position having the largest magnitude among $r_{m,n}$, $m = 0, 1, \dots, N - 1$. When successive data symbols are opposite in sign, using only the information contained in \mathbf{r}_n is not sufficient for acquisition unless data transition occurs at the 0th-lag position, a condition that does not occur frequently. To achieve rapid acquisition, we make use of the information provided by both \mathbf{r}_n and $\bar{\mathbf{r}}_n$. In case successive data symbols are opposite in sign, the correlation peak among $\bar{r}_{m,n}$, $m = 0, 1, \dots, N - 1$, is located where data transition occurs because of an intentional reversal of sign during correlation of the signal as seen from (2). Therefore, a data-modulated DSSS signal can be acquired by locating the time position having the largest magnitude among the elements of \mathbf{r}_n and $\bar{\mathbf{r}}_n$. Note that acquisition can be accomplished when \mathbf{r}_n and $\bar{\mathbf{r}}_n$ are available, that is, after N signal samples are obtained. On the other hand, acquisition using a parallel correlator requires a longer time of $2N - 1$ sampling periods.

Other applications wherein the all-lag reference-code correlator has an advantage over the parallel correlator include the following examples.

- In mobile communications, estimation of the impulse response of a multipath fading channel is often required at the receiver in order to enhance the system performance. When a parallel correlator is used, correlation results obtained at successive sampling instants constitute a channel estimate. Obtaining these correlation results involves more than N data samples. On the other hand,

the information of the whole channel estimate is contained in \mathbf{r}_n , wherein the peaks appeared in the elements of \mathbf{r}_n correspond to the multipaths. Only N data samples are involved. Thus, an all-lag correlator is faster than a parallel correlator in channel estimation. In addition, the channel estimate can be obtained more directly and more conveniently from a knowledge of \mathbf{r}_n . More frequent update of channel estimates is also made possible, which improves the receiver performance in response to rapidly varying channels.

- By processing the additional correlation information provided by an all-lag reference-code correlator, both acquisition and channel estimation can be made more robust to noise and interference than using a parallel correlator.
- Code tracking and automatic frequency control in DSSS receivers can be made easier and can be enhanced by more frequent adjustments.

III. RECURSIVE ALL-LAG REFERENCE-CODE CORRELATOR FOR GENERATING $\{\mathbf{r}_n\}$

Define an $N \times N$ shift matrix

$$\mathbf{S} = \begin{bmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & 0 & 0 & \dots & 0 & 0 \end{bmatrix}. \tag{9}$$

This shift matrix performs a linear transformation on a length- N column vector by cyclically shifting up the elements in the vector by one step. This transformation can be realized in practice by using an end-around shift register. Since cyclically shifting a length- N vector for N steps reproduces the original vector, it follows that

$$\mathbf{S}^N = \mathbf{I} \tag{10}$$

where \mathbf{I} is an $N \times N$ identity matrix.

The desired recursive form for \mathbf{r}_n is obtained by expressing \mathbf{r}_n in terms of \mathbf{r}_{n-1} . Let \mathbf{c}_m be the m th column¹ of \mathbf{C} where $m = 0, 1, \dots, N - 1$. That is,

$$\mathbf{c}_m = [c_m, c_{m-1}, \dots, c_1, c_0, c_{N-1}, c_{N-2}, \dots, c_{m+2}, c_{m+1}]^T. \tag{11}$$

It is easy to verify that

$$\begin{aligned}
 \mathbf{c}_{N-1} &= \mathbf{S} \mathbf{c}_0 \\
 \mathbf{c}_{m-1} &= \mathbf{S} \mathbf{c}_m, \quad m = 1, 2, \dots, N - 1. \tag{12}
 \end{aligned}$$

Since $\mathbf{C} = [\mathbf{c}_0 \ \mathbf{c}_1 \ \dots \ \mathbf{c}_{N-1}]$, it follows that (1) can be expressed as

$$\mathbf{r}_n = \sum_{m=0}^{N-1} d_{n+m-(N-1)} \mathbf{c}_m. \tag{13}$$

Setting $m' = m + 1$ in (13), we find that

$$\mathbf{r}_n = d_n \mathbf{c}_{N-1} + \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} \mathbf{c}_{m'-1}. \tag{14}$$

¹Throughout this paper, the left uppermost element of a matrix is assigned an index (0, 0) rather than the usual index (1, 1).

Applying (12) to this expression gives

$$\mathbf{r}_n = d_n \mathbf{c}_{N-1} + \mathbf{S} \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} \mathbf{c}_{m'} + d_{n-N} (\mathbf{S} \mathbf{c}_0 - \mathbf{c}_{N-1}). \quad (15)$$

Noting that $\mathbf{r}_{n-1} = \sum_{m'=0}^{N-1} d_{n-1+m'-(N-1)} \mathbf{c}_{m'}$ as seen from (13), we arrive at the desired recursive form

$$\mathbf{r}_n = \mathbf{S} \mathbf{r}_{n-1} + (d_n - d_{n-N}) \mathbf{c}_{N-1}. \quad (16)$$

Based on a knowledge of \mathbf{r}_{n-1} , one can generate \mathbf{r}_n by this recursive relationship. Note that $\mathbf{S} \mathbf{r}_{n-1}$ is an end-around rotation of \mathbf{r}_{n-1} . As (16) is a recursive equation only, it remains to find the initial condition that makes (16) and (1) yield the same result. Repeated application of (16) for N times followed by applying (10) and (12) gives

$$\mathbf{r}_n = \mathbf{r}_{n-N} + \mathbf{C}(\mathbf{d}_n - \mathbf{d}_{n-N}). \quad (17)$$

Without loss of generality, we assume that signal samples d_n s are only available for $n = 1, 2, 3, \dots$ and that it is desired to generate $\mathbf{r}_N, \mathbf{r}_{N+1}, \mathbf{r}_{N+2}, \dots$. If we provide an initial condition that $\mathbf{r}_0 = \mathbf{0}$ and $d_0 = d_{-1} = \dots = d_{-(N-1)} = 0$, then (17) becomes identical to (1) for $n = N$. Based on a valid result of \mathbf{r}_N , one can compute \mathbf{r}_n , $n > N$, by using (16). Note that intermediate results $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{N-1}$ are not valid.

The number of arithmetic operations of the recursion for each iteration can easily be observed to be one subtraction, N additions, and N multiplications. Based on the recursive relationship of (16), the recursive all-lag reference-code correlator that generates $\{\mathbf{r}_n\}$ can be constructed as depicted in Fig. 2. It requires a length- N shift register to store the input signal samples, and output storage to store the N correlation results for the previous sampling instant, a negator, N multipliers and $N + 1$ two-input adders. Notice that prior to operation, the values stored in the shift register and in the output storage are initialized to zero. Table I summarizes the required numbers of components for implementing this recursive all-lag reference-code correlator. The numbers of components for an all-lag correlator directly implemented by N parallel correlators are also listed for reference. It is apparent that the order of implementation complexity is reduced from N^2 to N when the recursive form is used. Comparing with corresponding numbers of components for realizing a parallel correlator as listed also in Table I, one immediately finds that implementation complexity of a recursive all-lag reference-code correlator that generates $\{\mathbf{r}_n\}$ is approximately the same as that of a conventional parallel correlator. In particular, the degrees of complexity of both correlators are of the order N .

IV. RECURSIVE ALL-LAG REFERENCE-CODE CORRELATOR FOR GENERATING $\{\bar{\mathbf{r}}_n\}$

We proceed to derive the recursive formula for computing $\bar{\mathbf{r}}_n$ based on the same steps as in deriving the one for \mathbf{r}_n in Section III. Define an $N \times N$ shift matrix

$$\bar{\mathbf{S}} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -1 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \quad (18)$$

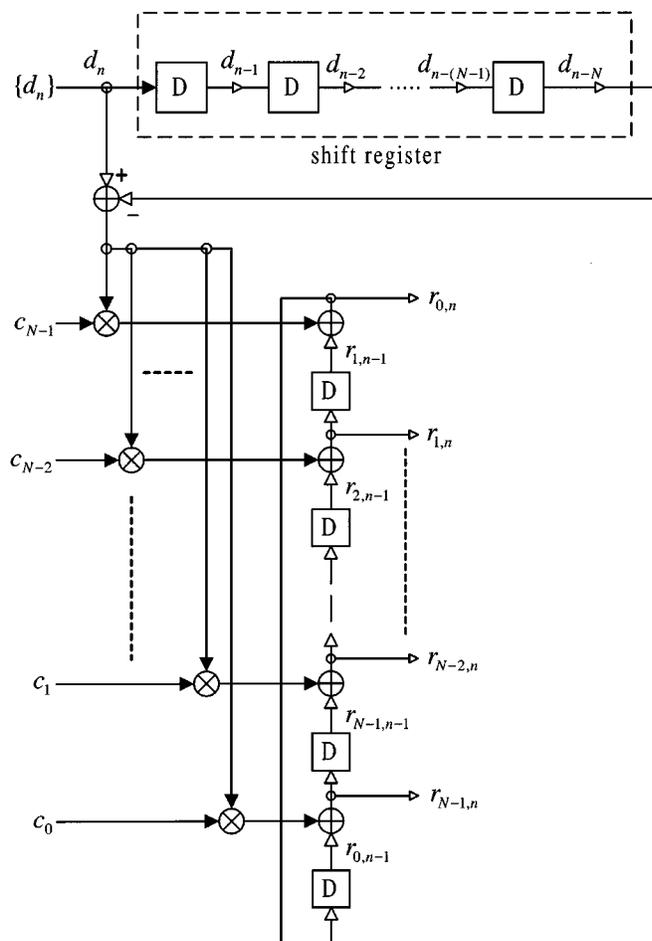


Fig. 2. Recursive all-lag reference-code correlator that generates a sequence of all-lag even-correlation vectors $\{\mathbf{r}_n\}$.

which performs a linear transform on a length- N column vector by cyclically shifting up the elements by one step and reversing the sign of the resultant lowest element. This transform is realized in practice by an inverting end-around shift register. It is easy to show that

$$\bar{\mathbf{S}}^N = -\mathbf{I}. \quad (19)$$

Let $\bar{\mathbf{c}}_m$ denote the m th column of $\bar{\mathbf{C}}$ for $m = 0, 1, \dots, N-1$, namely,

$$\bar{\mathbf{c}}_m = [c_m, c_{m-1}, \dots, c_1, c_0, -c_{N-1}, -c_{N-2}, \dots, -c_{m+2}, -c_{m+1}]^T. \quad (20)$$

It can be easily shown that

$$\begin{aligned} \bar{\mathbf{c}}_{N-1} &= -\bar{\mathbf{S}} \bar{\mathbf{c}}_0 \\ \bar{\mathbf{c}}_{m-1} &= \bar{\mathbf{S}} \bar{\mathbf{c}}_m, \quad m = 1, 2, \dots, N-1. \end{aligned} \quad (21)$$

Since $\bar{\mathbf{C}} = [\bar{\mathbf{c}}_0 \ \bar{\mathbf{c}}_1 \ \dots \ \bar{\mathbf{c}}_{N-1}]$, we can express (2) as

$$\bar{\mathbf{r}}_n = \sum_{m=0}^{N-1} d_{n+m-(N-1)} \bar{\mathbf{c}}_m \quad (22)$$

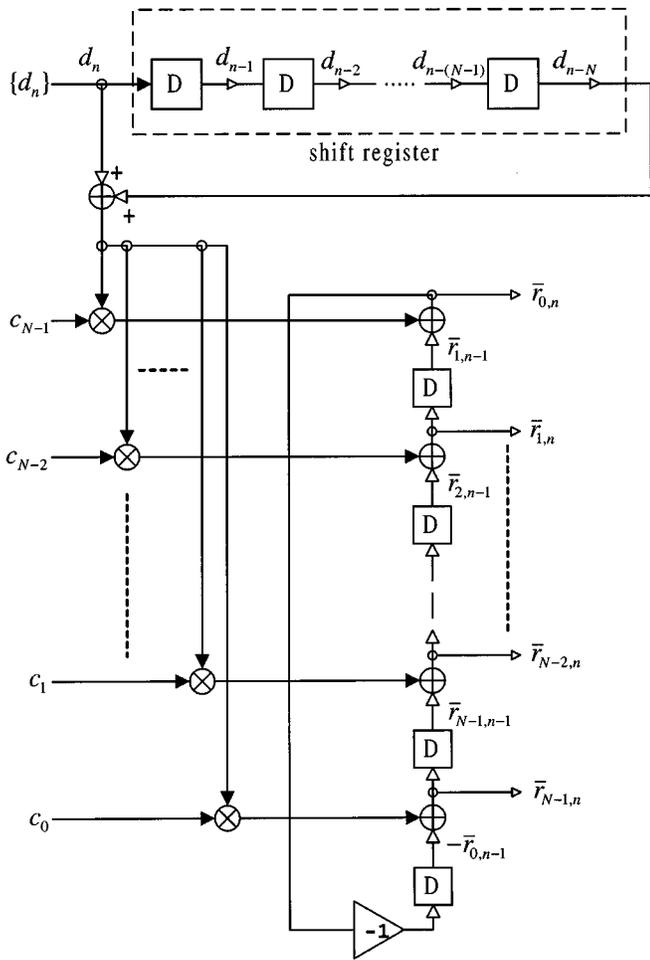


Fig. 3. Recursive all-lag reference-code correlator that generates a sequence of all-lag odd-correlation vectors $\{\bar{r}_n\}$.

so that

$$\bar{r}_n = d_n \bar{c}_{N-1} + \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} \bar{c}_{m'-1}. \quad (23)$$

Applying (21) to the last expression yields

$$\begin{aligned} \bar{r}_n &= d_n \bar{c}_{N-1} + \bar{S} \sum_{m'=1}^{N-1} d_{n-1+m'-(N-1)} \bar{c}_{m'} \\ &\quad + d_{n-N} (\bar{S} \bar{c}_0 + \bar{c}_{N-1}). \end{aligned} \quad (24)$$

It follows that the recursive relationship is given by

$$\bar{r}_n = \bar{S} \bar{r}_{n-1} + (d_n + d_{n-N}) \bar{c}_{N-1} \quad (25)$$

which enables generation of \bar{r}_n based on \bar{r}_{n-1} . Notice that $\bar{S} \bar{r}_{n-1}$ is an inverting end-around rotation of \bar{r}_{n-1} . The initial condition is derived as follows. Repeated application of (25) for N times followed by an application of (19) and (21) yields

$$\bar{r}_n = -\bar{r}_{n-N} + \bar{C}(d_n + d_{n-N}). \quad (26)$$

Again, assume that signal samples, d_n s, are only available for $n = 1, 2, 3, \dots$ and that we want to generate $\bar{r}_N, \bar{r}_{N+1}, \bar{r}_{N+2}, \dots$. It is easy to identify the desired initial condition to be $\bar{r}_0 = \mathbf{0}$ and $d_0 = d_{-1} =$

$\dots = d_{-(N-1)} = 0$, so that (26) and (2) become identical for $n = N$. The recursive formula (25) can be used thereafter to generate $\bar{r}_n, n > N$.

Fig. 3 shows a recursive all-lag reference-code correlator that generates $\{\bar{r}_n\}$ and that is constructed according to (25). It is apparent that implementation of this correlator requires a length- N shift register for storing the input signal sequence, N storage elements to retain the correlation results, a negator, N multipliers and $N + 1$ two-input adders. Again, values in the shift register and the output storage are reset to zero prior to operation. Table I lists the required numbers of components for realizing this recursive correlator, along with those results for other correlators. It is shown that a substantial reduction of implementation complexity is obtained when the recursive form, rather than the direct-implementation method, is employed. Results of Table I also indicate that this recursive correlator is of the order N in the implementation complexity, the same order as that of a parallel correlator.

V. CONCLUSION

Recursive forms for generating all-lag correlation sequences $\{r_n\}$ and $\{\bar{r}_n\}$ have been derived. It has been shown that using these recursive forms, all-lag reference-code correlators can be implemented with a complexity approximately the same as that of a conventional parallel correlator. Degrees of implementation complexity for recursive all-lag reference-code correlators have therefore been reduced substantially from order N^2 to order N .

REFERENCES

- [1] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [2] M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*, Revised ed. New York: McGraw-Hill, 1994.
- [3] M. B. Pursley, "Performance evaluation for phased-coded spread-spectrum multiple-access communication—Part I: System analysis," *IEEE Trans. Commun.*, vol. COMM-25, pp. 795–799, Aug. 1977.
- [4] D. V. Sarwate and M. B. Pursley, "Crosscorrelation properties of pseudorandom and related sequences," *Proc. IEEE*, vol. 68, pp. 593–619, May 1980.
- [5] H. T. Kung, "Why systolic architectures?," *Computer*, vol. 15, pp. 37–46, Jan. 1982.
- [6] D. T. Magill and G. Edwards, "Digital matched filter ASIC," in *Proc. IEEE MILCOM'90*, Sept. 30–Oct. 3, 1990, pp. 235–238.
- [7] R. S. Mowbray and P. M. Grant, "Simplified matched filter receiver designs for spread spectrum communications applications," *IEE Elect. Commun. Eng. J.*, pp. 59–64, Apr. 1993.
- [8] R. C. Dixon and J. S. Vanderpool, "Spread spectrum correlator," U.S. Patent 5 022 047, June 4, 1991.
- [9] —, "Dual-threshold spread spectrum correlator," U.S. Patent 5 719 900, Feb. 17, 1998.
- [10] R. Price and P. E. Green, Jr., "A communication technique for multipath channels," *Proc. IRE*, vol. 46, pp. 555–570, Mar. 1958.
- [11] M. Luise and R. Reggiannini, "Carrier recovery in all-digital modems for burst-mode transmission," *IEEE Trans. Commun.*, vol. 43, pp. 1169–1178, Feb. 1995.
- [12] U. Fawer, "A coherent spread-spectrum diversity receiver with AFC for multipath fading channels," *IEEE Trans. Commun.*, vol. 42, pp. 1300–1311, Feb. 1994.
- [13] A. Q. Hu, P. C. K. Kwok, and T. S. Ng, "MPSK DS/CDMA carrier recovery and tracking based on correlation technique," *Electron. Lett.*, vol. 35, pp. 201–203, Feb. 1999.
- [14] J. Zhang and Q. Zhang, "An approach to realize the parallel real-time correlator based on parallel pipeline from addition network," in *Proc. IEEE TENCON*, Beijing, China, Oct. 21, 1993, pp. 849–852.
- [15] W. C. Lin, K. C. Liu, and C. K. Wang, "Differential matched filter architecture for spread spectrum communication system," *Electron. Lett.*, vol. 32, pp. 1539–1540, Aug. 1996.
- [16] B. S. E. Tan and G. J. R. Povey, "Low complexity spread spectrum correlator," *Electron. Lett.*, vol. 33, pp. 1204–1205, July 1997.

- [17] S. H. Ahn, J. T. Kim, and Y. H. Lee, "Efficient implementation of parallel correlators for code acquisition in DS/CDMA systems," in *Proc. IEEE ISCAS*, May 20–June 2 1999, pp. IV-576–IV-579.
- [18] J. Li and S. Tantaratana, "Optimal and suboptimal coherent acquisition schemes for PN sequences with data modulation," *IEEE Trans. Commun.*, vol. 43, pp. 554–564, Feb. 1995.

A Systematic Approach in Constructing Fully Differential Amplifiers

Gonggui Xu and Sherif H. K. Embabi

Abstract—Based on constructing the Common Mode Feedback path to be topologically similar to the differential mode path, a systematic mapping approach for deriving a fully differential amplifier, from its single ended counterpart, is presented. The motivation, usage and efficiency of the proposed approach is demonstrated by two examples.

I. INTRODUCTION

In integrated circuit design, fully differential amplifiers are popular because they have better Power Supply Rejection Ratio (PSRR) than their single ended counterparts. For high gain fully differential amplifiers, an internal Common Mode Feedback (CMFB) path must be added to establish a common mode (i.e. average) output voltage over all working frequencies. Two tasks exist in the construction of a CMFB path: how to generate a CMFB control signal, and where to inject the CMFB control signal back to the biasing. The CMFB control signal can be generated either by a continuous-time approach or by a switched-capacitor approach, the detailed discussion can be found in [1]. In this paper, our discussion is restricted to the injection of the continuous-time CMFB control signal back into the differential mode path.

Some multi-stage fully differential amplifier topologies can be found in the literature [2], [3]. It's interesting to see that in both topologies, the CMFB control signal is injected back into the first stage. Is there any particular reason that the first stage is preferred over other stages? How is the CMFB path compensated? In this paper, we will try to answer above questions and a systematic approach for constructing fully differential amplifiers will be formulated.

The rest of the paper is organized as follows. In Section II, a two-stage Miller amplifier will be used as an example to present the motivation and procedure of the proposed approach. In Section III, a more complicated four-stage amplifier topology is used as another example to verify the efficiency of the proposed approach. The conclusion is given in Section IV.

II. MOTIVATION AND PROCEDURE

The two-stage Miller amplifier, shown in Fig. 1, will be considered to discuss the properties of fully differential amplifiers and the CMFB

path requirement. The CMFB control signal V_{ctrl} in Fig. 1 is injected back into the first stage of the two differential channels, as in [2], [3]. The reasons for that is discussed next.

When the circuit, shown in Fig. 1, operates in the differential mode, there is no common mode variation and the CMFB circuit can be ignored. The differential mode small signal model for one channel can be depicted as in Fig. 2, where g_{m1} and g_{m2} are transconductances of transistors M1(M2) and M9(M10) respectively. The CMFB path of Fig. 1 consists of two parts: from the average output V_{cm} (point a) to the feedback control signal V_{ctrl} (point b) and from V_{ctrl} (point b) to the two amplifier outputs (point c). The first part (from a to b) has much larger bandwidth and smaller DC gain (approximately -1 if properly designed). Most of the CMFB frequency characteristics is determined by the second part (from b to c). The single channel small signal model of the second part is shown in Fig. 3 where g_{mc} and g_{m2} are transconductances of transistors M3(M4) and M9(M10), respectively.

Comparing Fig. 3 with Fig. 2, one can see that two small signal models share the same compensation capacitor C_m and g_{m2} stage, hence are topologically similar. This is an important observation. The similarity of the topologies leads to a stable CMFB path if the differential mode path is stable. This can be explained by their transfer functions which are given by:

$$\frac{V_{out}}{V_{ctrl}} = \frac{-g_{mc}C_m s + g_{mc}g_{m2}}{s^2 C_m C_L + s C_m g_{m2} + g_{o1}g_{o2}} \quad (1)$$

$$\frac{V_{out}}{V_{in}} = \frac{-g_{m1}C_m s + g_{m1}g_{m2}}{s^2 C_m C_L + s C_m g_{m2} + g_{o1}g_{o2}} \quad (2)$$

where, g_{o1} and g_{o2} are total output conductances at node V_1 and V_{out} , respectively. Notice that above two equations are very similar, actually the denominators are the same. Equation (1) shows that a high CMFB gain ($g_{mc}g_{m2}/g_{o1}g_{o2}$) and a large CMFB bandwidth (g_{mc}/C_m) are achieved.

The gain and bandwidth requirements for CMFB paths depend on the amplifier's common mode/power supply gain and bandwidth. For example, in the circuit shown in Fig. 1, a differential pair is used in the input stage and hence the circuit's common mode has a small gain (approximately $g_{ob}g_{m2}/g_{o1}g_{o2}$) and a small bandwidth (approximately g_{ob}/C_m), where g_{ob} is the output conductance of bias current source transistors M5 & M6; also the noise from this circuit's positive power supply is only amplified by a small gain (approximately $\Delta g_m g_{m2}/g_{o1}g_{o2}$) and it has a small bandwidth (approximately $\Delta g_m/C_m$), where Δg_m is the g_m process mismatch between load transistor M3 and M4. Therefore, the corresponding CMFB path gain and bandwidth can be small. But, on the other hand, high CMFB path gain gives more accurate common mode bias and large CMFB bandwidth improves the PSRR at high frequencies. Therefore, a high CMFB gain and a large CMFB bandwidth are always preferred whenever they are achievable (sometimes they come for free and can be well-compensated when the differential mode path is shared). This is the case in the circuit of Fig. 1 and it has a high CMFB gain and a large CMFB bandwidth which is achieved by injecting the CMFB control signal back to the first stage.

The above discussion applies not only to the two-stage Miller amplifier but also to the more general fully differential amplifiers including the multi-stage amplifiers in [2], [3]. This explains why in all of these fully differential amplifiers, the CMFB control signal is injected back into the first stage.

The concept of sharing and topology similarity also allow for developing a systematic approach to construct fully differential amplifiers. It will be demonstrated that the circuit shown in Fig. 1 can be derived

Manuscript received March 14, 1999; revised August 2000. This work was supported in part by Semiconductor Research Corporation. This paper was recommended by Associate Editor G. de Veirman.

The authors were with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA. They are now with Texas Instrument Incorporated, Dallas, TX 75243 USA.

Publisher Item Identifier S 1057-7130(00)01135-6.