Transactions Briefs

Novel Parallel Architectures for Short-Time Fourier Transform

K. J. Ray Liu

***Abstract*—Novel parallel architectures for short-time Fourier transform based on adaptive time-recursive processing is proposed for efficient VLSI implementation. Only $N-1$ multipliers and $N+1$ adders are required. The proposed approach can be easily extended to multi-dimensional cases without the transpose operation. Various properties of the proposed architectures are also presented.**

I. Introduction

The short-time Fourier transform (STFT) has played a significant role in digital signal processing, including speech, music, and radar/sonar applications [1], [5], [6]. Due to the demand of high throughput of these applications, efficient parallel architectures that enable real-time implementation of the STFT are quite essential [3], [4].

The expression for the discrete-time STFT is given by

$$X(n_0, \omega) = \sum_{n=-\infty}^{\infty} x(n)w(n_0 - n)e^{-j\omega n} \tag{1}$$

where $w(n)$ is an analysis window. At each time instant, the STFT is a discrete-time Fourier transform or discrete Fourier transform (DFT). Many DFT computing algorithms and architectures have been proposed. However, the direct use of the DFT without considering the special *sliding window* effect of STFT is not an efficient approach. There are two major approaches for STFT. One is based on the filter bank approach and another is the FFT-based approach [1].

The filter bank approach can be described by the following two convolution sum equations,

$$X(n, \omega_0) = [x(n)e^{-j\omega_0 n}] * w(n) \tag{2}$$

or

$$X(n, \omega_0) = e^{-j\omega_0 n}[x(n) * w(n)e^{-j\omega_0 n}] \tag{3}$$

where $*$ denotes the convolution sum. Equations (2) and (3) can be implemented as DF Formatted Filterbanks [1]. It can be seen that each channel of the filter bank requires a convolution sum that needs $O(N^2)$ operations. If the throughput rate is N, then N multipliers are required. Accordingly, the total number of multipliers required for the STFT system is on the order of $O(N^2)$.

The FFT-based approach is well known [2]. The major disadvantage is the need of global interconnections in the butterfly computations. This is a disaster in VLSI implementation, especially when hundreds (or even thousands) of channels are required in the applications. Besides, the total number of multipliers is on the order of $O(N \log N)$.

Manuscript received January 23, 1992; revised November 30, 1992, and May 18, 1993. This work was supported in part by the Office of Naval Research under Grant N00014-93-1-0566 and the NSF grant MIP9309506. This paper was recommended by Associate Editor K. Yao.

The author is with the Department of Electrical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742.

IEEE Log Number 9210911.

Recent advancement of VLSI/ULSI technologies has made it practical to build low-cost and high-density application-specific integrated circuits (ASIC) to meet the demands of speed and performance of signal processing. In this paper, novel parallel architectures for STFT are proposed for efficient VLSI implementation.

II. The Novel Architectures

As mentioned before, the STFT is mainly used in real-time applications where new data keep arriving so that the on-line computation is definitely essential. Note that the high complexity in the existing approaches results from the direct computation of each newly windowed data. Old information is not adequatedly used to reduce the computational complexity. To reduce the complexity, the concept of adaptive processing can be exploited, especially in the applications where new data keep arriving.

For simplicity, let us assume a rectangular window first. Suppose

$$X(n_0, \omega) = \sum_{n=n_0}^{n_0+N-1} x(n)e^{-j\omega(n-n_0)}$$

has been obtained, the relation between $X(n_0+1, \omega)$ and $X(n_0, \omega)$ can be shown to be

$$\begin{aligned} X(n_0+1, \omega) &= \sum_{n=n_0+1}^{n_0+N} x(n)e^{-j\omega(n-n_0-1)} \\ &= e^{j\omega}[X(n_0, \omega) - x(n_0) \\ &\quad + x(n_0+N)e^{-j\omega N}]. \end{aligned} \tag{4}$$

If the discrete Fourier transform (DFT) is considered, i.e., $\omega_k = 2\pi k/N$, $k = 0, 1, \cdots, N-1$, then

$$X(n_0+1, k) = e^{j(2\pi k/N)}[X(n_0, k) + (x(n_0+N) - x(n_0))], \quad k = 0, 1, \cdots, N-1. \tag{5}$$

The architecture for the above equation is given in Fig. 1. The total number of multipliers required for the STFT system is $N-1$ (since the first channel does not need one) and the number of adders is $N+1$. The throughput rate is 1 to obtain $X(n_0+1, k)$ from $X(n_0, k)$ and the throughput rate of obtaining nonoverlapped windowed STFT is N. A downsampling can be added as shown in Fig. 2 if it is used as a DFT-based analysis filter bank which found lots of application is subband coding and transform-domain filtering [3], [4].

For the two-dimensional (2-D) STFT, suppose the STFT of $X(m_0, n_0, k, l)$ is available and the window is moving along the m direction as shown in Fig. 3, the update relation between $X(m_0+1, n_0, k, l)$ and $X(m_0, n_0, k, l)$ can be obtained as

$$\begin{aligned} X(m_0+1, n_0, k, l) = [X(m_0, n_0, k, l) + \sum_{n=0}^{N-1} (x(m_0+N, n) \\ - x(m_0, n))e^{-j(2\pi nl/N)}]e^{j(2\pi k/N)}. \end{aligned} \tag{6}$$

Here a 1-D STFT (with rectangular window) of the error term $\Delta x = x(m_0+N, n) - x(m_0, n)$ needs to be computed first. The architecture is given in Fig. 4 which contains a 1-D architecture given in Fig. 1 to perform the 1-D STFT and an update loop to obtain the new 2-D transform. There are N linear arrays of size N to store the columns ($l = 0, 1, \cdots, N-1$) of $X(m_0, n_0, k, l)$. The computation

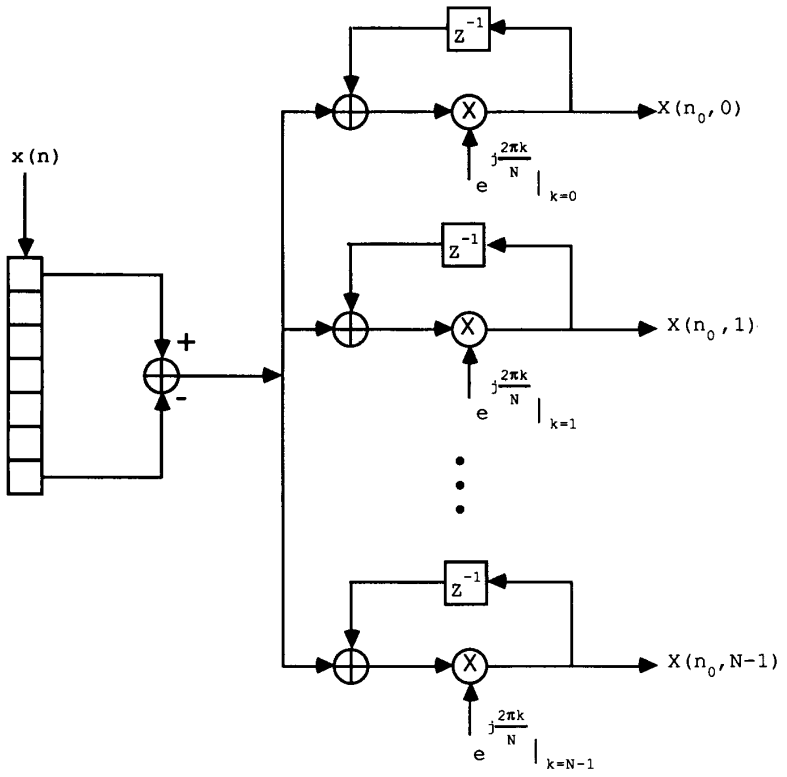

Fig. 1. New architecture for short-time Fourier transform.

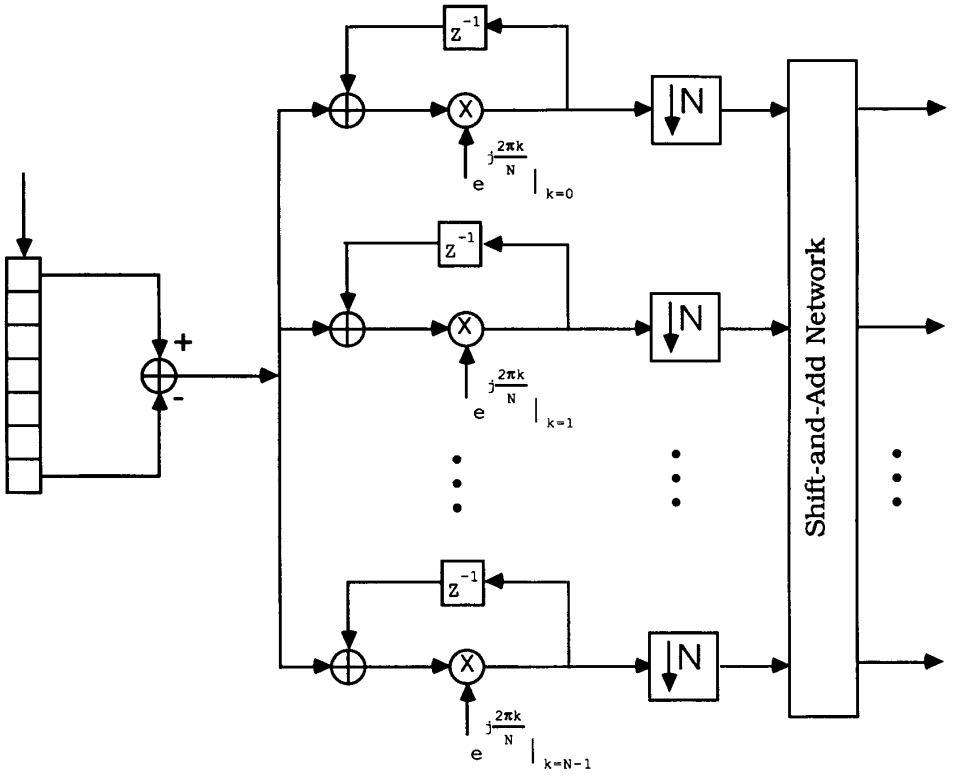

Fig. 2. Application to multirate filter bank and windowing obtained from the shift-add-add network.

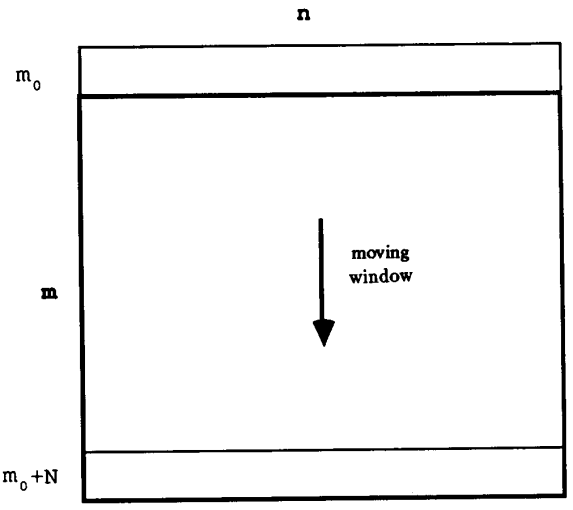

Fig. 3. Moving window of 2-D short-time Fourier transform.

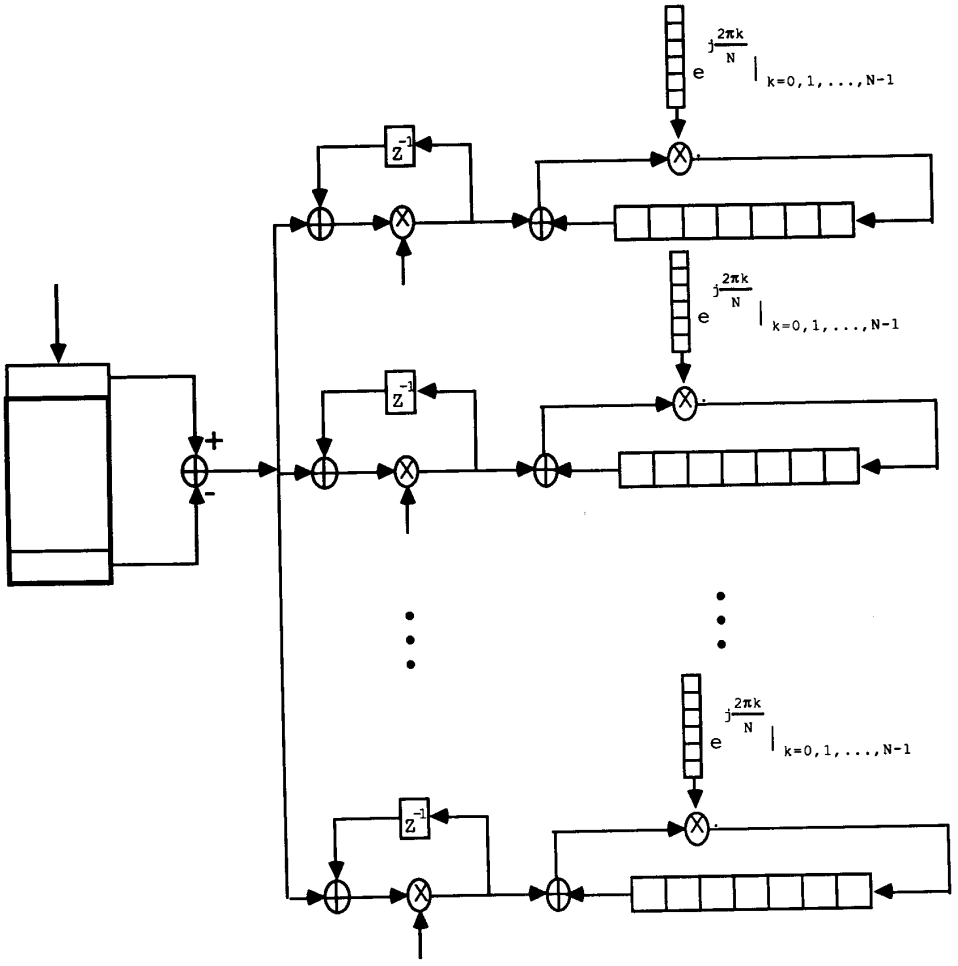

Fig. 4. Parallel architecture for 2-D short-time Fourier transform.

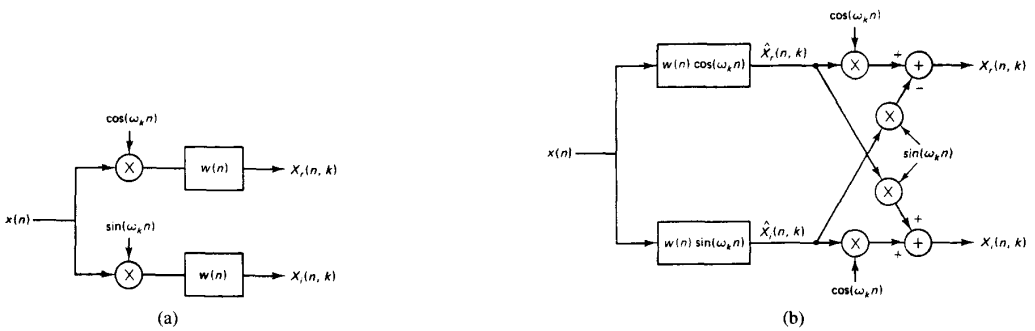

Fig. 5. Implementation of (2) and (3) using real operations.

of each column spectrum is independent of each other. An interesting property of this approach is that no transpose operation is required. The total number of multipliers is $2N-1$ and that of adders is $2N+1$. The throughput rate is $2N$ for the 2-D STFT.

In general, for a M-D STFT, the update equation is given by

$$X(\underline{m}_t, \underline{k}) = [X(\underline{m}_{t-1}, \underline{k}) + \boldsymbol{F}_{M-1}(\Delta x(\underline{m-1}))] \cdot e^{j(2\pi k/N)} \quad (7)$$

where $\boldsymbol{F}_{M-1}$ denotes an $(M-1)$-D STFT. The computational structure from $(M-1)$-D to M-D STFT is the same as in Fig. 4.

Equations (2), (3), and (4) involve the multiplication of a complex exponential factor. If real operations are to be used, the implementations using (2) and (3) are obtained in Fig. 6(a) and Fig. 6(b), respectively [1]. Obviously, each complex channel becomes two real channels. However, in our approach as shown in Fig. 1 using (4), a CORDIC processor [10] is enough to handle the case. To see this, denote the input to the multiplier in Fig. 1 as $a_r + ja_i$ and the output as $b_r + jb_i$, we have

$$\begin{bmatrix} b_r \\ b_i \end{bmatrix} = \begin{bmatrix} \cos\frac{2\pi k}{N} & -\sin\frac{2\pi k}{N} \\ \sin\frac{2\pi k}{N} & \cos\frac{2\pi k}{N} \end{bmatrix} \begin{bmatrix} a_r \\ a_i \end{bmatrix}. \quad (8)$$

This is a simple planer rotation which can be easily carried out by using CORDIC processor without explicitly performing the multiplication.

This approach can also be viewed as a STFT IIR filtering. The filter, as shown in Fig. 1, consists of two parts: a FIR section and an IIR section. It is because of the IIR part that simplifies the computational structure of the STFT which is basically a FIR system in nature. The transfer function of the IIR section is given by

$$H_k(z) = \frac{e^{j(2\pi k/N)}}{1 - e^{j(2\pi k/N)} z^{-1}}, \qquad k = 0, 1, \cdots, N-1. \quad (9)$$

The poles are on the unit circle for all channels. This may cause the instability. Fortunately, the poles are at $\cos\theta_k + j\sin\theta_k$, $k = 0, 1, \cdots, N-1$. Hence, by quantizing the coefficients $\cos\theta_k$ and $\sin\theta_k$, we directively quantize the real and imaginary parts of the poles so that they can be always guaranteed to locate inside the unit circle. Such phenomenon is similar to that of the normal-form structure of an IIR filter [9].

A disadvantage of the proposed architecture is that not all the well-known windows are applicable. Equation (5) serves as the most fundamental formula for this approach with the assumption of rectangular window. If it is not a rectangular window, then there is not as simple as update equation as (5). For windows such as the Hanning window given by $w_H(n) = \frac{1}{2}[1 - \cos(2\pi n/N)]$, the relation of the spectrum of the windowed and nonwindowed (rectangular window) data is $X_H(k) = -\frac{1}{4}X(k-1) + \frac{1}{2}X(k) - \frac{1}{4}X(k+1)$. Only shift-and-add operation is needed to modified from $X(k)$ to $X_H(k)$ without explicitly performing the multiplication, where $X(k)$ is the rectangular-windowed spectrum that can be easily obtained. There are many such kinds of windows that provide excellent performance with only shift-and-add operation required to be modified from $X(k)$ [8]. A shift-and-add network can be added as shown in Fig. 2 to obtain the windowed STFT. The shift-and add network is locally interconnected and regular so that it is not a problem for VLSI implementation.

III. Conclusions

The proposed architectures are very efficient in term of hardware complexity and throughput rate. In particular, since the hardware complexity is on the order of N and $2N$ for 1-D and 2-D STFT, respectively, the architectures are very suited for VLSI implementation of STFT with large number of channels. The approach is also applicable to multirate signal processing, especially the DFT-based filter bank.

References

[1] S. H. Nawab and T. F. Quatieri, "Short-time Fourier transform," in Lim and Oppenheim (Ed.), *Advanced Topics in Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1988.

[2] G. B. Adams *et al.*, "Experiments with parallel fast Fourier transforms," in Bayoumi (Ed.), *Parallel Algorithms and Architectures for DSP Applications*, Kluwer Academic Publishers, 1991.

[3] S. D. Silverstein, W. E. Engeler, and J. A. Tardif, "Parallel architectures for multirate superresolution spectrum analyzers," *IEEE Trans. Circuits Syst.*, vol. 38, no. 4, pp. 449–453, Apr. 1991.

[4] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Mag.*, pp. 10–37, Jan. 1992.

[5] M. R. Portnoff, "Representation of digital signals and systems based on the short-time Fourier transform," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 28, pp. 55–69, Feb. 1980.

[6] D. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech, Signal Process.*, vol. 32, pp. 236–243, Apr. 1984.

[7] J. L. Aravena, "Recursive moving window DFT algorithm," *IEEE Trans. Comput.*, vol. 39, pp. 145–148, Jan. 1990.

[8] F. C. Marshall and G. C. Temes, "Binary windows for the discrete Fourier transform," *Proc. IEEE*, vol. 63, pp. 1370–1371, Sept. 1975.

[9] L. B. Jackson, *Digital Filters and Signal Processing*, 2nd ed., Kluwer Academic Publishers, 1989.

[10] J. Volder, "The CORDIC trigonomitric computing technique," *IRE Trans. Electron. Comput.*, vol. 8, pp. 330–334, Sept. 1959.

[11] K. J. R. Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms," *IEEE Trans. Signal Process.*, vol. 41, no. 3, pp. 1357–1377, Mar. 1993.

Analog Multiplierless LMS Adaptive FIR Filter Structures

Huang Qiuting and George S. Moschytz

***Abstract*— Two adaptive FIR filter configurations are proposed for implementing the LMS algorithm with no along delay elements or multipliers. The first uses a linear feedback shift register (LFSR) to generate pseudorandom binary sequences for applications where white noise has been traditionally used. The binary signals are delayed by a shift register and multiplied by on/off switching so that the resulting structure is free of analog delay lines and multipliers. For applications where inputs are colored, the second configuration uses a Σ–Δ modulator as front-end for converting the adaptive filter's input into a binary sequence. Such a filter tends to adapt itself into a low-pass characteristic to remove the high frequency noise due to the modulator.**

I. Introduction

Adaptive signal processing has many applications such as system modeling, identification, adaptive control, equalization, and interference canceling [1, 2]. The ease with which digital circuits can be programmed has made the implementation of adaptive algorithms an essentially digital subject. In recent years, however, there is growing interest in implementing various adaptive algorithms in analog form, in order to take advantage of the lower power consumption, smaller silicon area, and higher speed of monolithic analog circuits [3]–[7]. The least mean-square (LMS) algorithm is the most commonly used because it is easiest to implement in hardware. Analog configurations have been proposed for both recursive (IIR) and nonrecursive (FIR) LMS algorithms [3], [4]. Like their digital counterparts, the IIR implementation of the LMS algorithm is efficient but its stability is difficult to control. By contrast, an LMS FIR filter is intrinsically stable because its error surface is quadratic, and convergence therefore guaranteed. The latter is therefore more popular despite its higher cost of implementation.

When an analog LMS adaptive FIR filter takes on essentially the same configuration as its digital counterpart [3], it requires an analog delay line, multipliers, integrators, and a summer circuit. In terms of cost, at least one analog delay element consisting of an op-amp and a few capacitors, one integrator, and two multipliers are required per filter tap-coefficient. The corresponding silicon area restricts the number of taps that can be integrated on a single chip to a very small number, which limits its range of useful applications. The analog components also introduce various imperfections such as noise, power-supply coupled interference, clock feedthrough, offset, and component mismatch. While the component mismatch can be corrected by the adaptive mechanism of the filter itself, the interferences, which accumulate from all the taps, contribute to the excessive mean-squared error (MSE) of the filter. Analog adaptive FIR filters therefore still have some way to go before their implementation becomes feasible in monolithic form. It is toward this goal, however, that this paper explores new structures that are more feasible as adaptive filters in analog integrated circuit form.

Manuscript received June 12, 1992; revised February 17, 1993. This paper was recommended by Associate Editor D. J. Allstot.

The authors are with the Institute for Signal and Information Processing, Swiss Federal Institute of Technology, ETH-Zentrum, 8092 Zurich, Switzerland.

IEEE Log Number 9210908.

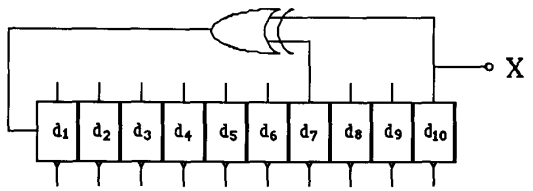

Fig. 1. A 10-bit maximal-length LFSR pseudo-random sequence generator.

II. Multiplierless LMS Configuration for System Modeling and Identification

In order to stimulate an unknown "plant" with an impulse (i.e., sine waves at all frequencies), white noise is typically used as the signal source for applications such as system modeling and identification. In digital adaptive filters elaborate pseudorandom number (PN) generators are used to generate such inputs [1], [2]. However, storing and propagating white noise in multibit digital or analog form and multiplying it using multibit digital or analog multipliers is highly inefficient compared to binary PN sequences which can be generated by very simple digital circuits. The latter have autocorrelation and spectral characteristics closely approximating those of true white noise [8], [9]. Fig 1 shows such a binary PN sequence generator based on the maximal length linear feedback shift register (LFSR). Binary PN sequences are sometimes found in adaptive filter applications such as in communications systems, but their use seems more dictated by communications requirements such as minimizing multipath effects, etc., than a clear realization that binary PN signals can be used in any adaptive filter where white noise is required at the input.

A switched-capacitor (SC) based multiplierless adaptive filter configuration with a binary delay line is shown Fig. 2. The binary input signal X to the adaptive filter may be a pseudorandom sequence specially generated by an LFSR or it may happen to be binary code words in certain applications, such as adaptive path identification in spread-spectrum communications [1] or binary signals representing the sign of the input to an echo cancellor [6]. In Fig. 2 the binary sequence is shifted down a binary delay line whose outputs $d_1, \cdots, d_i, \cdots, d_n$ control whether the plus or minus error signal e is added to the corresponding weight-integrators. They also control whether plus or minus integrator output w_i, representing the ith weight (tap-coefficient), is added to the SC summer circuit. The integrator symbol in Fig. 2 designates the SC circuit shown in Fig. 3, which contains additional features compared to a standard SC integrator for offset and clock feedthrough compensation [10]. The filter coefficients are represented in the form of analog voltage w_i at the op-amp output in Fig. 3. The overall algorithm implemented by Fig. 2 is

$$e(k) = Y(k) - w_{01}(k-1) - \sum_{i=1}^{n} w_i(k-1)d_i(k),$$
$$\mathrm{d}_i(k) = \pm 1 \quad (1)$$

where Y is the "plant" output shown in Fig. 2. The weight update