TECHNICAL RESEARCH REPORT

S Y S T E M S
R E S E A R C H
C E N T E R

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Novel Parallel Architectures for Short-Time Fourier Transform

by K.J.R. Liu

TR 92-4

Novel Parallel Architectures for Short-Time Fourier Transform

K.J. Ray Liu
Electrical Engineering Department
Systems Research Center
University of Maryland, College Park, MD 20742

Abstract

In this paper, 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.

1 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 as the convolution sum. Equations (2) and (3) can be implemented as given in Fig.1.a and Fig.1.b, respectively [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 $O(N \log N)$.

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.

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

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

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

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

The architecture for the above equation is given in Fig.2. The total number of multipliers required for the STFT system is $N-1$ (since the first channel does not need one) and the numer 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 non-overlapped windowed STFT is N. A downsampling can be added as shown in Fig.3 if it is used as a DFT-based analysis filter bank which found lots of application in 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.4, the update relation between $X(m_0+1, n_0, k, l)$ and $X(m_0, n_0, k, l)$ can be obtained as

$$X(m_0+1, n_0, k, l) = [X(m_0, n_0, k, l) + \sum_{n=0}^{N-1}(x(m_0+N, n) - x(m_0, n))e^{-j\frac{2\pi nl}{N}}] \cdot e^{j\frac{2\pi k}{N}}. \quad (6)$$

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.5 which contains a 1-D architecture given in Fig.2 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 of each column spectrum is independent to 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

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

where $\mathbf{F}_{M-1}$ denotes as a $(M-1)$-D STFT. The computational structure from $(M-1)$-D to M-D STFT is the same as in Fig.5.

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) as given in Fig.1 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.2 using (4), a CORDIC processor [10] is enough to handle the case. To see this, denote the input to the multiplier in Fig.2 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.2, 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

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

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 no as simple update equation as (5). For windows such as the Hanning window given by $w_H(n) = \frac{1}{2}[1 - \cos\frac{2\pi n}{N}]$, the relation of the spectrum of the windowed and non-windowed (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 can be easily obtained. There are many such kinds of windows that provide excellent performance with only shift-and-add operation required to modified from $X(k)$ [8]. A shift-and-add network can be added as shown in Fig.3 to obtain the windowed STFT. The shift-and-add network is locally interconnected and regular so that is not a problem for VLSI implementation.

3 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 Advanced Topics in Signal Processing, Ed. Lim and Oppenheim, Prentice-Hall, 1988.

[2] G.B. Adams et al, "Experiments with parallel fast Fourier transforms", in Parallel Algorithms and Architectures for DSP Applications, Ed. Bayoumi, 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 and Systems, Vol 38, No 4, pp.449-453, April 1991.

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

[5] M.R. Portnoff, "Representation of digital signals and systems based on the short-time Fourier transform", IEEE Trans. ASSP, Vol 28, pp.55-69, Feb. 1980.

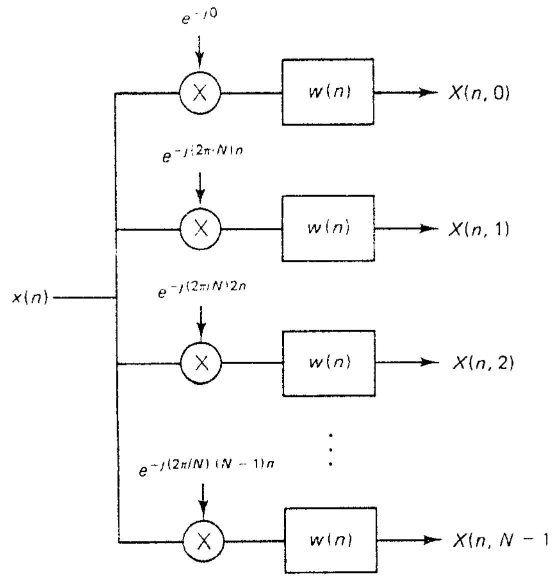
[6] D. Griffin and J.S. Lim, "Signal estimation from modified short-time Fourier transform", IEEE Trans. ASSP, Vol 32, pp.236-243, April 1984.

[7] J.L. Aravena, "Recursive moving window DFT algorithm", IEEE Trans. on Computer, Vol 39, pp.145-148, Jan. 1990.

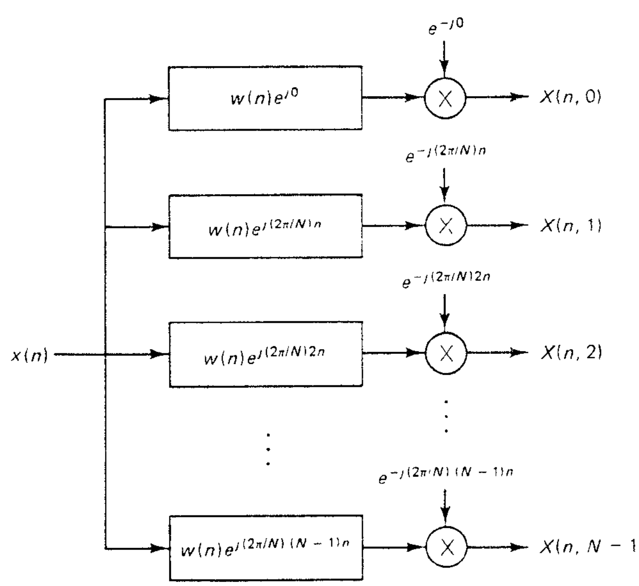
[8] F.C. Marshall and G.C. Temes, "Binary windows for the discrete Fourier transform", Proc. IEEE, Vol 63, pp.1370-1371, Sep. 1975.

[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, Sep. 1959.



(a)



(b)

Figure 1 Filter bank realization of short-time Fourier transform
(a) Realization of equation (2), (b) realization of (3)

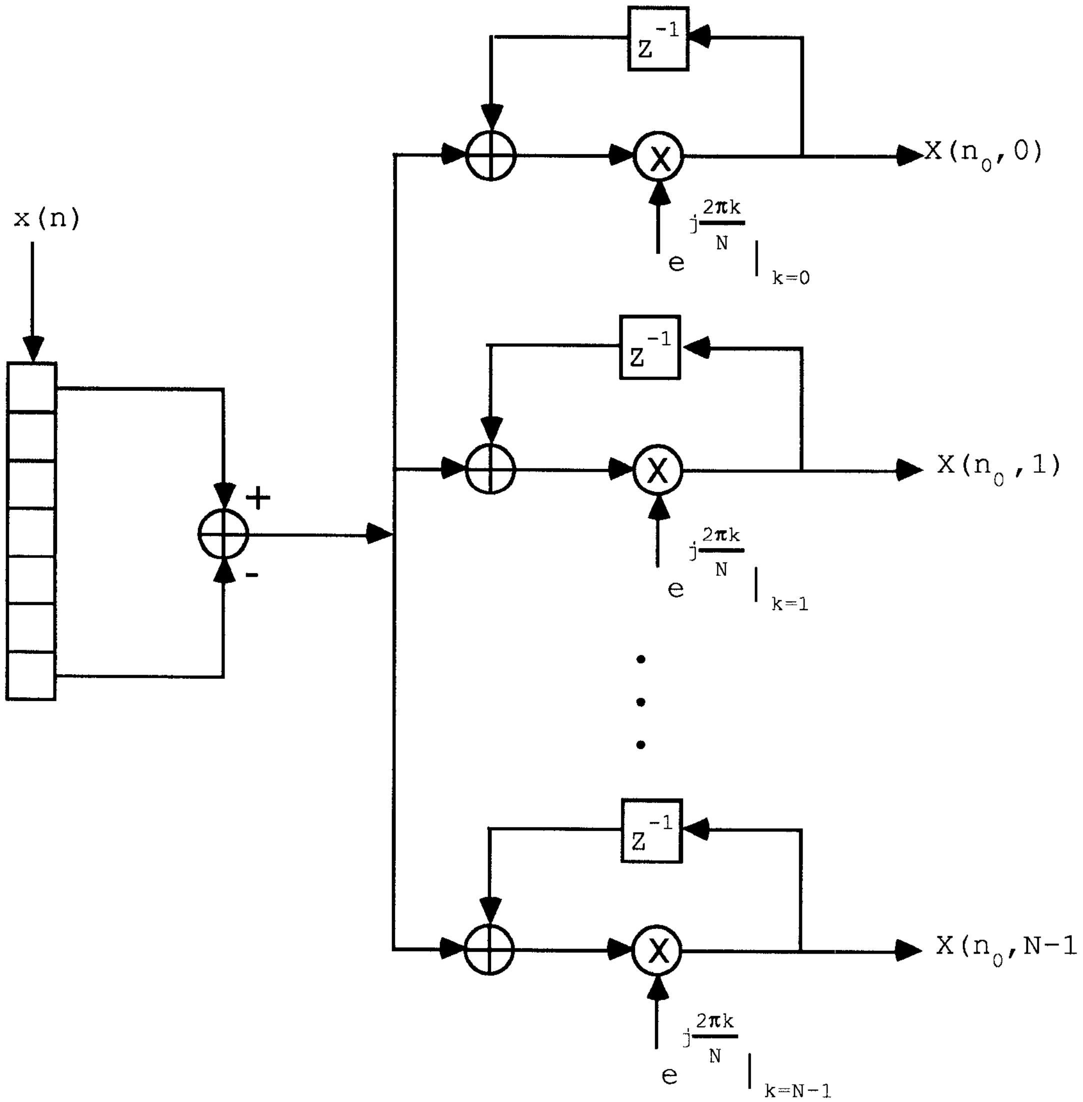


Figure 2 The new architecture for short-time Fourier transform

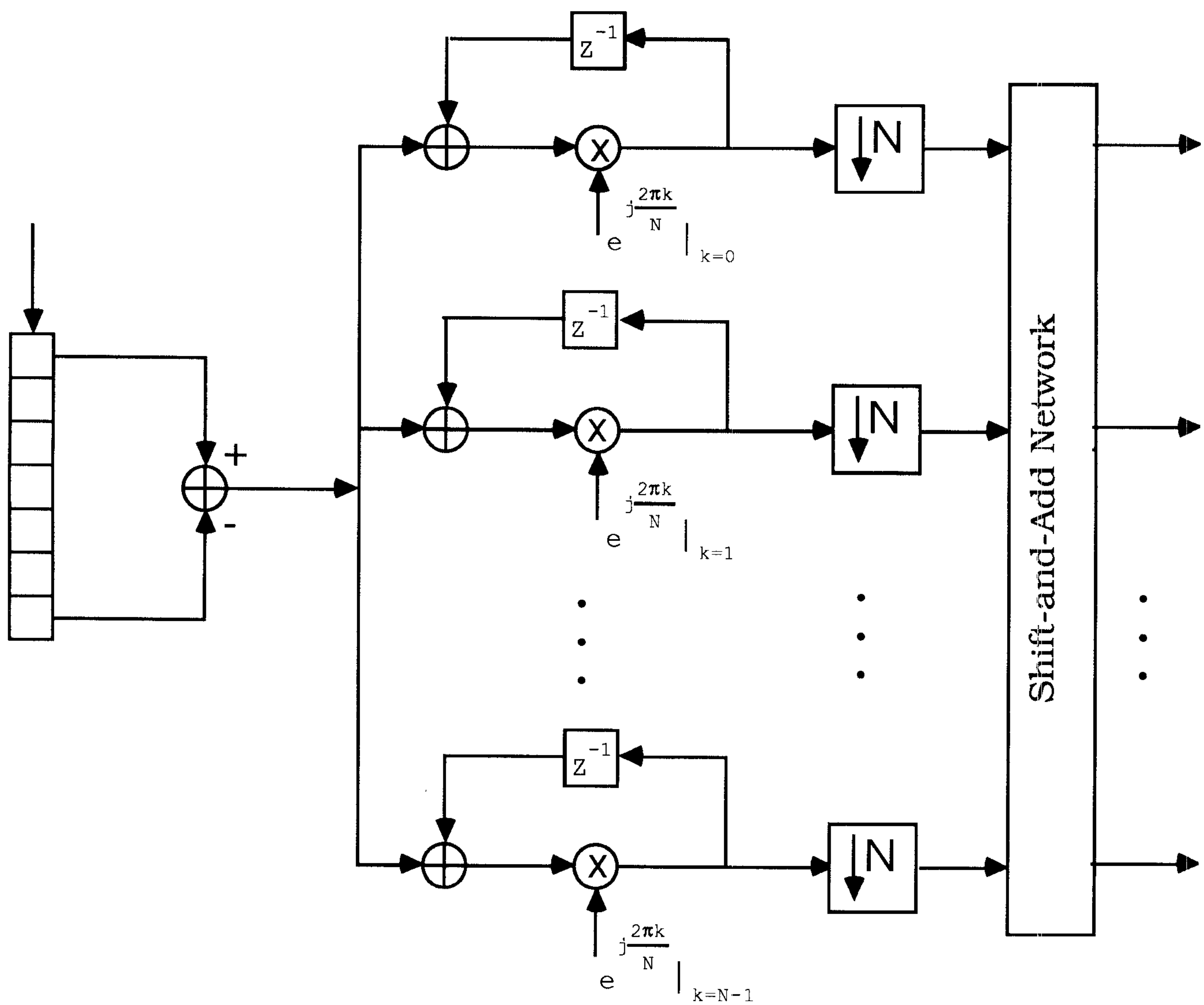


Figure 3 Application to multirate filter bank and windowing obtained from the shift-add-add netowrk

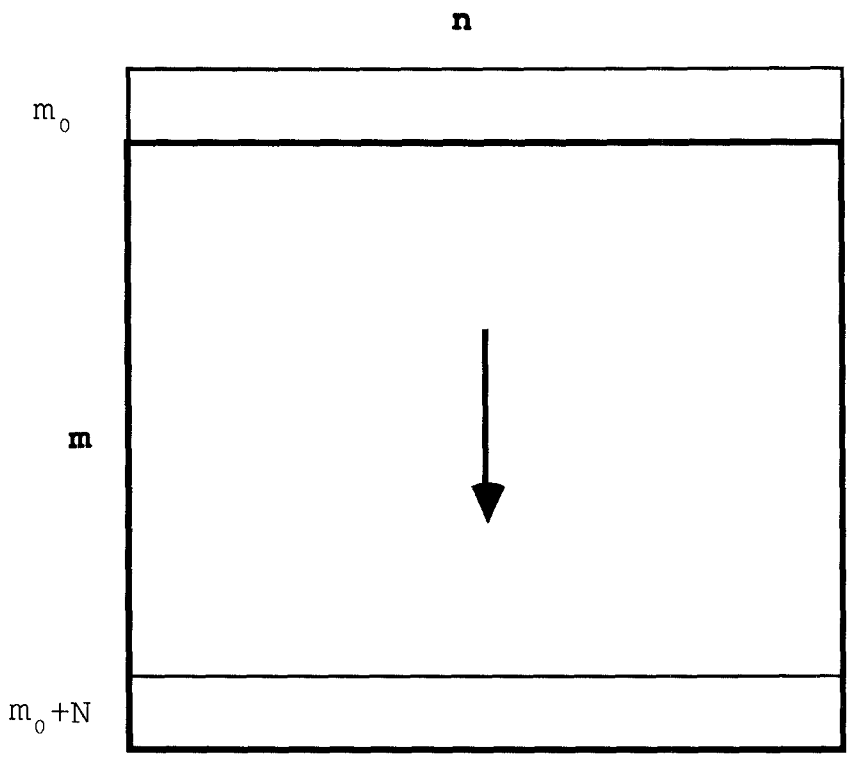


Figure 4 The moving window of 2-D short-time Fourier transform

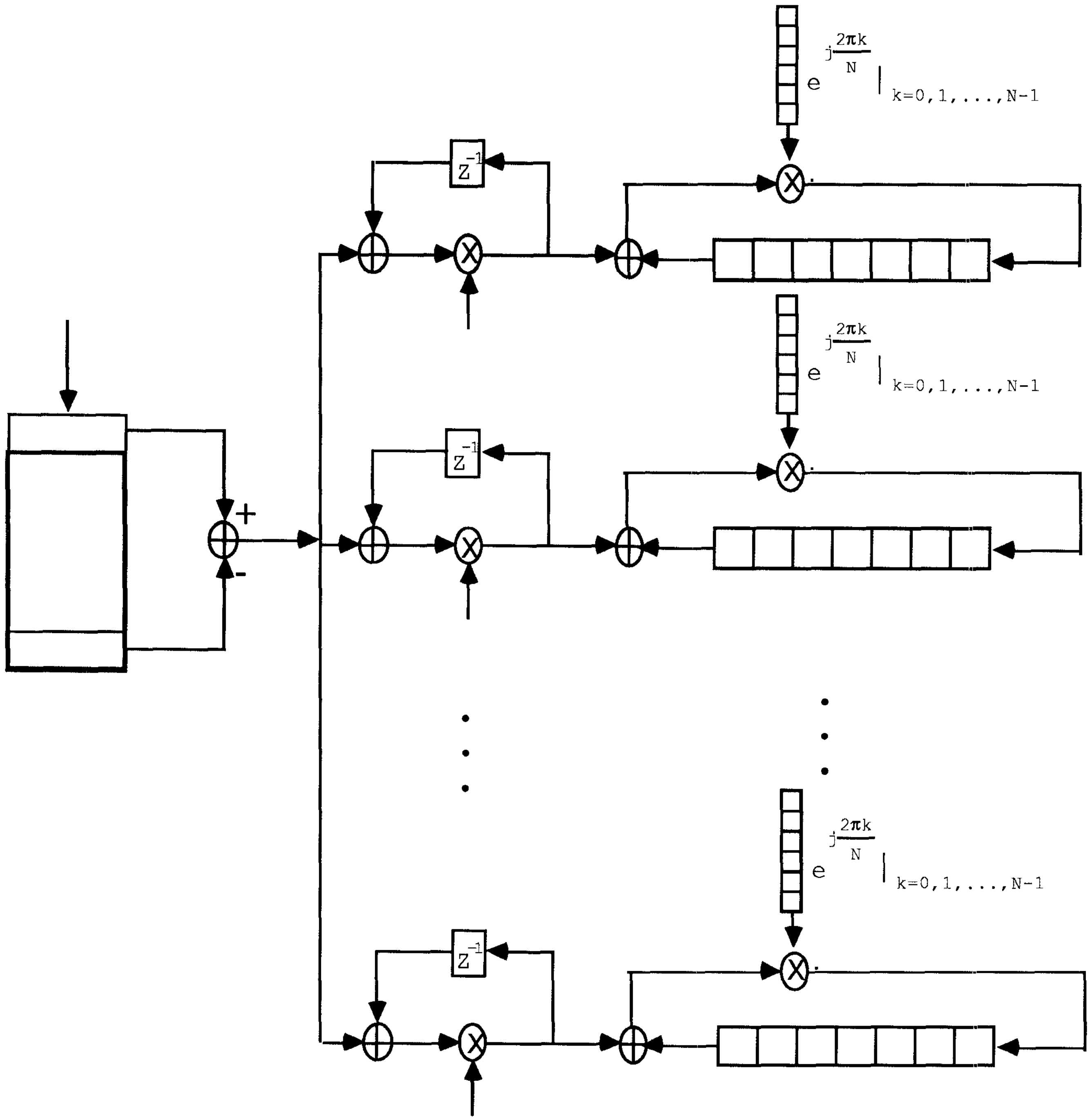
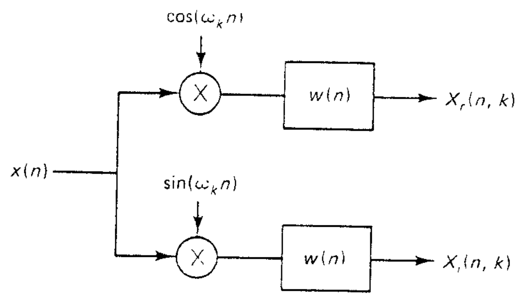
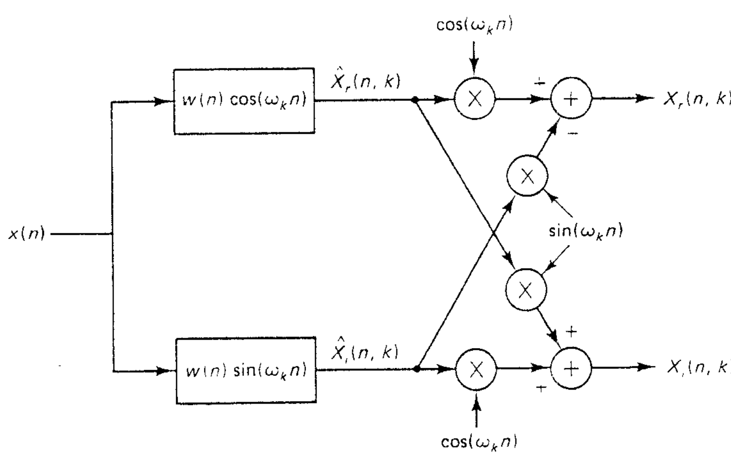


Figure 5 The parallel architecture for 2-D short-time Fourier transform



(a)



(b)

Figure 6 Implementation of equation (2) and (3) using real operations