TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

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

Optimal Unified Architectures for the Real-Time Computation of Time-Recursive Discrete Sinusoidal Transforms

by K.J.R. Liu, C.T. Chiu, R.K. Kolagotla, and J.F. JaJa

TR 92-66

Optimal Unified Architectures for the Real-Time Computation of Time-Recursive Discrete Sinusoidal Transforms

K.J.R. Liu, C.T. Chiu, R. K. Kolagotla, and J. F. JáJá

Electrical Engineering Department
Systems Research Center
University of Maryland
College Park, MD 20742
Ph: (301) 405-6619

ABSTRACT

An optimal unified architecture that can efficiently compute the Discrete Cosine, Sine, Hartley, Fourier, Lapped Orthogonal, and Complex Lapped transforms for a continuous input data stream is proposed. This structure uses only half as many multipliers as the previous best known scheme [1]. This architecture is regular, modular, and has only local interconnections in both the data and control paths. There is no limitation on the transform size N and only $2N-2$ multipliers are needed for the DCT. The throughput of this scheme is one input sample per clock cycle. We provide a theoretical justification by showing that any discrete transform whose basis functions satisfy the *Fundamental Recurrence Formula* has a second-order autoregressive structure in its filter realization. We also demonstrate that dual generation transform pairs share the same autoregressive structure. We extend these time-recursive concepts to multi-dimensional transforms. The resulting multi-dimensional structures are fully-pipelined and consist of only d 1-D transform arrays and shift registers, where d is the dimension.

This work was partially supported by NSF grant ECD-8803012.

1 Introduction

Discrete sinusoidal transforms play significant roles in various digital signal processing applications, such as spectrum analysis, image and speech signal processing, computer tomography, data compression, and signal reconstruction[2, 3, 4, 5]. Among different discrete sinusoidal transforms, the discrete cosine transform (DCT)[6, 7, 8, 9], the discrete sine transform (DST)[10, 9], the discrete Hartley transform (DHT)[27, 28, 25, 6], and the discrete Fourier transform (DFT)[26, 3] are widely used because of their efficient performance[5, 18, 19, 21, 22]. Recently, the Lapped Orthogonal Transform (LOT)[14], and the Complex Lapped transform (CLT)[13] were introduced for transform coding with significantly reduced blocking effects and for motion estimation.

In real-time signal processing applications, especially in speech/image communications and radar/sonar signal processing, input data arrive serially. In traditional FFT based algorithms, these serial data are buffered and then transformed using the FFT scheme of complexity $O(N \log N)$[3]. Buffering the serial data requires $O(N)$ time. In this paper, we describe a novel architecture that merges the buffering and transform operations into a single unit of total hardware complexity $O(N)$. Unlike the FFT, this architecture has only local interconnections and is better suited for VLSI implementations. It is important to note that the proposed architectures generate time-recursive transforms, not just block transforms, *i.e.*, the transform of the N points $[x(t+1), x(t+2), \ldots, x(t+N)]$ is generated one clock cycle after the transform of $[x(t), x(t+1), \ldots, x(t+N-1)]$ is generated. To generate time-recursive transforms, the traditional fast algorithms based architectures require $O(logN)$ time using $O(N \log N)$ hardware, while the architectures we propose require only a constant time with $O(N)$ hardware. Time-recursive transforms are currently gaining widespread use in motion estimation, in video signal processing, and in reducing blocking effects in data compression.

We have shown in [1] that when discrete transforms are performed on segments of a continuously incoming data stream, transforms can be realized by a unified lattice structure with a data throughput rate of one input sample per clock cycle. This architecture is regular, modular, and free of global interconnections. Unlike the many fast algorithms for DFT, DCT, and DHT, there is no constraint on the transform size N. Table 1 [1] summarizes a comparison of the time-recursive approach with other well-known fast algorithms. A time-recursive lattice 2-D DCT structure with

applications to the HDTV systems is also given in [16]. This 2-D DCT structure requires only two 1-D DCT blocks and is fully-pipelined with no transposition.

In this paper, we describe an optimal unified filter structure, which preserves the advantages of the lattice architecture, while reducing the hardware complexity in half. In the time-recursive lattice architecture, two transforms called the dual generated pairs, are obtained simultaneously. The unified filter structure is more suitable for applications where only one transform is required. We develop a systematic approach to derive the time-recursive unified filter architectures for any discrete transform. We show that all the resulting unified filter architectures have a similar second-order autoregressive structure with minimum number of multipliers. A theoretical basis for this fact is provided. We also demonstrate that the time-recursive concept can be generalized to multi-dimensional transforms by using only the one-dimensional transform architecture and simple shift registers. An area-time complexity analysis is also provided to show that the proposed approach is asymptotically optimal in speed and area.

The rest of this paper is organized as follows. The unified lattice structure for sinusoidal transforms is described in Section 2. The derivation for obtaining the optimal unified filter structure from the transfer function of the discrete sinusoidal transforms is discussed in Section 3. The architectures of the inverse discrete sinusoidal transforms based on the IIR filter realization are presented in Section 4. In Section 5, the characteristics of these architectures are discussed from a theoretical point of view. The unified architectures for time-recursive based multi-dimensional discrete sinusoidal transforms are discussed in Section 6. Finally we give a conclusion in Section 7.

2 Lattice Structure for Discrete Sinusoidal Transforms

The time-recursive approach has been shown to be efficient in both hardware and computational complexity for the computation of discrete sinusoidal transforms (DXT), (such as the DCT, DST, and DHT), for time series input data stream [1]. In this section, we will extend to the DFT, LOT and CLT and provide a unified view of lattice structures for time-recursive approach.

Denote the discrete sinusoidal transform DXT of a data sequence of length N $[x(t), x(t +$

$1), ..., x(t+N-1); t = 0, 1, 2, ...]$ at time t as

$$X_X(k,t) = C(k) \sum_{n=t}^{t+N-1} x(n) P_{n-t}(k), k = 0, 1, ..., N-1, \tag{1}$$

where $P_{n-t}(k)$ are transform basis functions and $C(k)$ are constants used for normalization. It was shown in [1] that most discrete sinusoidal transforms have dual generated pairs. That is, the lattice structure used for generating one transform automatically generates its dual. For example, the dual of the DCT is the DST. Both the transform and its dual have similar updating relations. Let us denote the dual generated pairs by $X_{xc}(k,t)$ and $X_{xs}(k,t)$. Then, the time-recursive relation between $X_x(k,t)$ and $X_x(k,t+1)$ can be obtained by eliminating the effect of the first term of the previous sequence and updating the effect of the last term of the current sequence. In general, the dual generation properties between the transform pairs $X_{xc}(k,t)$ and $X_{xs}(k,t)$ are given by the following equations [1]:

$$\begin{aligned} X_{xc}(k,t+1) &= e(k)\{\left[X_{xc}(k,t) + [x(t+N)(-1)^k - x(t)]D_c\right]\Gamma_c \\ &\quad + \left[X_{xs}(k,t) + [x(t+N)(-1)^k - x(t)]D_s\right]\Gamma_s\} \end{aligned} \tag{2}$$

and

$$\begin{aligned} X_{xs}(k,t+1) &= f(k)\{\left[X_{xs}(k,t) + [x(t+N)(-1)^k - x(t)]D_s\right]\Gamma_c \\ &\quad - \left[X_{xc}(k,t) + [x(t+N)(-1)^k - x(t)]D_c\right]\Gamma_s\}, \end{aligned} \tag{3}$$

where D_c and D_s are the associated cos and sin transform kernels of the DXT with fixed index n. Coefficients $e(k)$ and $f(k)$ depend on the definition of the transforms and are always equal to one except for the two transforms LOT and CLT. That is, two transforms can be dually generated from each other in a lattice form as shown in Fig. 1. Here, we will briefly describe the definition of the various discrete sinusoidal transforms [10, 11, 12, 13] and state the equations corresponds to (2) and (3).

The one-dimensional (1-D) DCT of an input data sequence $[x(t), x(t+1), ..., x(t+N-1)$, t=0,1,2,...] is defined as[11]

$$X_c(k,t) = C(k)\sqrt{\frac{2}{N}} \sum_{n=t}^{t+N-1} x(n) \cos\left[(n-t+\frac{1}{2})\frac{k\pi}{N}\right], \qquad k = 0, 1, ..., N-1, \tag{4}$$

and the 1-D DST is defined as[10]

$$X_s(k,t) = C(k)\sqrt{\frac{2}{N}}\sum_{n=t}^{t+N-1} x(n)\sin\left[(n-t+\frac{1}{2})\frac{k\pi}{N}\right], \qquad k = 1,...,N, \tag{5}$$

where

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0 \text{ or } N\ , \\ 1 & \text{otherwise.} \end{cases}$$

The updated DCT and DST of the input data $[x(t), x(t+1), ..., x(t+N)]$ can be generated from $X_c(k,t)$ and $X_s(k,t)$ by following relations:

$$\begin{aligned} X_c(k,t+1) &= C(k)\sqrt{\frac{2}{N}}\sum_{n=t+1}^{t+N} x(n)\cos\left[\frac{\pi[2(n-t-1)+1]k}{2N}\right], \\ &= \{X_c(k,t) + \left[-x(t) + (-1)^k x(t+N)\right] C(k)\left(\sqrt{\frac{2}{N}}\right)\cos\left(\frac{\pi k}{2N}\right)\}\cos\left(\frac{\pi k}{N}\right) \\ &+\{X_s(k,t) + \left[-x(t) + (-1)^k x(t+N)\right] C(k)\left(\sqrt{\frac{2}{N}}\right)\sin\left(\frac{\pi k}{2N}\right)\}\sin\left(\frac{\pi k}{N}\right) \end{aligned} \tag{6}$$

and

$$\begin{aligned} X_s(k,t+1) &= \{X_s(k,t) + \left[-x(t) + (-1)^k x(t+N)\right] C(k)\left(\sqrt{\frac{2}{N}}\right)\sin\left(\frac{\pi k}{2N}\right)\}\cos\left(\frac{\pi k}{N}\right) \\ &\{X_c(k,t) + \left[-x(t) + (-1)^k x(t+N)\right] C(k)\left(\sqrt{\frac{2}{N}}\right)\cos\left(\frac{\pi k}{2N}\right)\}\sin\left(\frac{\pi k}{N}\right). \end{aligned} \tag{7}$$

That is, the DCT and DST can be dually generated time-recursively in a lattice form.

The definition of DHT is given by[12]

$$\begin{aligned} X_h(k,t) &= \frac{1}{\sqrt{N}}\sum_{n=t}^{t+N-1} x(n) cas\left(2(n-t)\frac{\pi k}{N}\right) \\ & k = 0,1,...,N-1, \end{aligned} \tag{8}$$

where $cas\theta \triangleq \cos\theta + \sin\theta$.

The Discrete Fourier Transform (DFT) of N samples $[x(t), x(t+1), \ldots, x(t+N-1)]$ is defined as[3]

$$X_f(k,t) = \frac{1}{\sqrt{N}}\sum_{n=t}^{t+N-1} x(n)\exp\{-j2(n-t)\frac{\pi k}{N}\}, \quad k = 0,1,\ldots,N-1. \tag{9}$$

The DHT uses real expressions $\cos(\frac{2\pi k(n-t)}{N}) + \sin(\frac{2\pi k(n-t)}{N})$ as the transform kernel, while discrete Fourier transform (DFT) uses the complex exponential expression $\exp(\frac{i2\pi k(n-t)}{N})$ as the transform kernel. Because the kernel of the DHT is a summation of cosine and sine terms, we can separate them into a combination of a DCT-like and a DST-like transforms as follows:

$$X_h(k,t) = \acute{X}_c(k,t) + \acute{X}_s(k,t), \tag{10}$$

where

$$\acute{X}_c(k,t) = \frac{1}{\sqrt{N}} \sum_{n=t}^{t+N-1} x(n) \left[\cos\left(\frac{2\pi k(n-t)}{N}\right)\right], \tag{11}$$

and

$$\acute{X}_s(k,t) = \frac{1}{\sqrt{N}} \sum_{n=t}^{t+N-1} x(n) \left[\sin\left(\frac{2\pi k(n-t)}{N}\right)\right]. \tag{12}$$

The $\acute{X}_c(k,t)$ is the so-called DCT-I and the $\acute{X}_s(k,t)$ is the DST-I that are defined by Yip and Rao in [5]. The $\acute{X}_c(k,t)$ and $\acute{X}_s(k,t)$ can be dually generated from each other by (2) and (3) with the corresponding coefficients listed in Table 2. Therefore, the DCT-I and DST-I are dual pairs. From the lattice structure shown in Fig. 1, the DHT are obtained by adding the the dual pairs $\acute{X}_c(k,t)$ and $\acute{X}_s(k,t)$. The real part of the DFT is $\acute{X}_c(k,t)$ and and the imaginary part is $\acute{X}_s(k,t)$.

The Complex Lapped Transform (CLT) [13] of $2N$ samples $[x(t-N+\frac{1}{2}), x(t-N+\frac{3}{2}), \ldots, x(x+N-\frac{1}{2})]$ is defined as

$$\begin{aligned} X_{clt}(k,t) &= \frac{1}{\sqrt{N}} \sum_{n=t-(N-\frac{1}{2})}^{t+(N-\frac{1}{2})} x(n) \exp\{-j\frac{(2k+1)(n-t)\pi}{2N}\} \cos\frac{(n-t)\pi}{2N}, \\ & \qquad k = 0, 1, \ldots, N-1. \end{aligned} \tag{13}$$

The Lapped Orthogonal Transform (LOT) [14, 13] of $2N$ samples $[x(t-N+\frac{1}{2}), x(t-N+\frac{3}{2}), \ldots, x(t+N-\frac{1}{2})]$ is defined as

$$X_{lot}(k,t) = \begin{cases} \sqrt{\frac{2}{N}} \sum_{n=t-(N-\frac{1}{2})}^{t+(N-\frac{1}{2})} x(n) \cos\frac{(2k+1)(n-t)\pi}{2N} \cos\frac{(n-t)\pi}{2N} + \alpha_k, \\ \qquad k = 0, 2, \ldots, (N-2), \text{even part of the CLT} \\ \sqrt{\frac{2}{N}} \sum_{n=t-(N-\frac{1}{2})}^{t+(N-\frac{1}{2})} x(n) \sin\frac{(2k+1)(n-t)\pi}{2N} \cos\frac{(n-t)\pi}{2N} + \beta_{nk}, \\ \qquad k = 1, 3, \ldots, (N-1), \text{odd part of the CLT} \end{cases} \tag{14}$$

where $\alpha_k = \beta_{nk} = 0$, except for $\alpha_0 = -(\sqrt{2}-1)/(2\sqrt{2})$, and $\beta_{n(N-1)} = (-1)^{n+\frac{1}{2}}\alpha_0$.
Since the LOT is obatined form the even and odd value of k in the CLT. We focus on the discussion of the dual generation for the CLT only. Define an Auxilary Complex Lapped Transform (ACLT) of $2N$ samples $[x(t-N+\frac{1}{2}), x(t-N+\frac{3}{2}), \ldots, x(x+N-\frac{1}{2})]$ as

$$X_{clt}(k,t) = \frac{1}{\sqrt{N}} \sum_{n=t-(N-\frac{1}{2})}^{t+(N-\frac{1}{2})} x(n) \exp\{-j\frac{(2k+1)(n-t)\pi}{2N}\} \sin\frac{(n-t)\pi}{2N},$$
$$k = 0, 1, \ldots, N-1. \tag{15}$$

Then, the CLT and ACLT can be dually generated from (2) and (3) with the corresponding coefficents listed in Table 2. All the transforms mentioned above can be realized by a lattice structure as shown in Fig. 1. This lattice structure is a modified normal form digital filter. Table 2 lists the coefficients in the unified lattice structure for different transforms. Here θ_k associated with the LOT/CLT equals $\frac{(2k+1)\pi}{4N}$.

3 Optimal Time-Recursive Architectures

3.1 Transfer Function Approach

Input data arrive serially in most real-time signal processing applications. If we can view the transform operation as a linear shift invariant (LSI) system which transforms the input sequence of samples into their transform coefficients, then it is similar to a filtering operation. The general approach to tackle a digital filter problem is to look at its transfer function. The transfer functions of the DXT can be derived using several approaches. We will derive them from the unified time-recursive lattice structures as shown in Fig. 1. The time difference equations [1] for the dually generated pairs are

$$y_{xc,k}(t) = e(k)\left\{\Gamma_c\left[D_c\tilde{x}(t) + y_{xc,k}(t-1)\right] + \Gamma_s\left[D_s\tilde{x}(t) + y_{xs,k}(t-1)\right]\right\} \tag{16}$$

and

$$y_{xs,k}(t) = f(k)\left\{\Gamma_c\left[D_s\tilde{x}(t) + y_{xs,k}(t-1)\right] - \Gamma_s\left[D_c\tilde{x}(t) + y_{xc,k}(t-1)\right]\right\}, \tag{17}$$

[1] The time index t is an integer parameter.

where

$$\tilde{x}(t) = (-1)^k x(t+N) - x(t), \tag{18}$$

and $y_{xc,k}(t)$ and $y_{xs,k}(t)$ corresponds to $X_{xc}(k,t)$ and $X_{xs}(k,t)$ in (2) and (3). The z transform of the above difference equations are

$$Y_{xc,k}(z) = e(k)\left\{(D_c\Gamma_c + D_s\Gamma_s)\,\tilde{X}(z) + \Gamma_c Y_{xc,k}(z)z^{-1} + \Gamma_s Y_{xs,k}(z)z^{-1}\right\} \tag{19}$$

and

$$Y_{xs,k}(z) = f(k)\left\{(D_s\Gamma_c - D_c\Gamma_s)\,\tilde{X}(z) + \Gamma_c Y_{xs,k}(z)z^{-1} - \Gamma_s Y_{xc,k}(z)z^{-1}\right\}. \tag{20}$$

$Y_{xs,k}(z)$ can be expressed in terms of $Y_{xc,k}(z)$ and $\tilde{X}(z)$ as

$$Y_{xs,k}(z) = \frac{f(k)\{(D_s\Gamma_c - D_c\Gamma_s)\,\tilde{X}(z) - \Gamma_s Y_{xc,k}(z)z^{-1}\}}{1 - f(k)\Gamma_c z^{-1}}, \tag{21}$$

it follows that the transfer function for $\frac{Y_{c,k}(z)}{X(z)}$ is

$$H_{xc,k}(z) = \frac{\left((-1)^k - z^{-N}\right)\left(e(k)[D_c\Gamma_c + D_s\Gamma_s] - e(k)f(k)D_c z^{-1}\right)}{1 - (e(k)+f(k))\Gamma_c z^{-1} + e(k)f(k)z^{-2}}. \tag{22}$$

Similarly, the transfer function for $\frac{Y_{s,k}(Z)}{X(z)}$ is

$$H_{xs,k}(z) = \frac{\left((-1)^k - z^{-N}\right)\left(f(k)[D_s\Gamma_c - D_c\Gamma_s] - e(k)f(k)D_s z^{-1}\right)}{1 - (e(k)+f(k))\Gamma_c z^{-1} + e(k)f(k)z^{-2}}. \tag{23}$$

From Table 2 and the transfer functions derived above, the transfer functions of different discrete sinusoidal transforms are given by

$$H_c(z) = \sqrt{\frac{2}{N}}C(k)\left((-1)^k - z^{-N}\right)\left(\cos\frac{\pi k}{2N}\right)\frac{\left(1 - z^{-1}\right)}{\left(1 - 2\left(\cos\frac{\pi k}{N}\right)z^{-1} + z^{-2}\right)}, \tag{24}$$

$$H_s(z) = -\sqrt{\frac{2}{N}}C(k)\left((-1)^k - z^{-N}\right)\left(\sin\frac{\pi k}{2N}\right)\frac{\left(1 + z^{-1}\right)}{\left(1 - 2\left(\cos\frac{\pi k}{N}\right)z^{-1} + z^{-2}\right)}, \tag{25}$$

$$H_h(z) = \frac{1}{\sqrt{N}}\left(1 - z^{-N}\right)\left(\frac{\cos\frac{2\pi k}{N} - \sin\frac{2\pi k}{N} - z^{-1}}{1 - 2\left(\cos\frac{2\pi k}{N}\right)z^{-1} + z^{-2}}\right), \tag{26}$$

$$H_f(z) = \frac{1}{\sqrt{N}}\left(1 - z^{-N}\right)\left(\frac{\cos\frac{2\pi k}{N} + j\sin\frac{2\pi k}{N} - z^{-1}}{1 - 2\left(\cos\frac{2\pi k}{N}\right)z^{-1} + z^{-2}}\right). \tag{27}$$

Because the size of the input data is $2N$ in the CLT, the updating vector is $1 - z^{2N}$ instead of $1 - z^{N}$. The transfer fucntion is obtained by substituting the corresponding coefficents in Table 2 to (16), which is

$$\begin{aligned} H_{clt}(z) &= \left(1 - z^{-2N}\right) \frac{1}{\sqrt{N}} j(-1)^{k+1} \frac{\left(\sin \frac{\pi}{4N}\right) e^{j\theta} \left(1 + e^{j2\theta} z^{-1}\right)}{1 - e^{j2\theta} 2 \left(\cos \frac{\pi}{2N}\right) z^{-1} + e^{j4\theta} z^{-2}}, \\ \theta &= \frac{(2k+1)\pi}{4N}. \end{aligned} \tag{28}$$

It follows that for the LOT,

$$H_{lote}(z) = evenpart\{H_{clt}\} \tag{29}$$

$$H_{loto}(z) = oddpart\{H_{clt}\}. \tag{30}$$

As we know from (1) that the transfer functions of these transforms are of finite impulse response. That is, the poles in the demoninator will be cancelled by the zeros of $((-1)^k - z^{-N})$ in the nominator. We observe that when the updating vector $(1 - z^{-N})$ is factored out, the basic structure of all the transforms is composed of a FIR and an IIR filter with a second order denominator and a first order numerator, *i.e.* we are using an IIR filter to realize a FIR filter. This realization can greatly reduce the hardware complexity compared with the implementation by FIR structures.

3.2 The Unified IIR Filter Architectures

From the transfer functions derived above, we observe that the DXT can be realized using a single universal filter module consisting of a shift register array and a second order IIR filter. This structure is depicted in Fig. 2. The coefficients of the universal IIR module for different transforms are listed in Table 3.

We note from (24) and (25) that the DCT and the DST share the same denominator and can be simultaneously generated using an IIR filter structure with three multipliers as depicted in Fig. 3. Compared with the lattice structure for the DCT and DST[1], the IIR realization requires only half as many multipliers. The difference is that the IIR structure implements the denominator of the transfer function in the direct form, while the lattice structure implements the poles in the normal form. From (26) and (27), we also observe that a single unified filter structure can be used to generate both the DHT and the DFT. This structure is depicted in Fig. 4.

The transfer function derived in (28) is in complex form. The IIR filter architecture for the LOT and the CLT is shown in Fig. 5. We will show in the following how to realize the CLT using real operations. The defintion of the CLT in (13) can be rewritten as

$$X_{clt}(k) = (-1)^k j \frac{1}{\sqrt{N}} \sum_{n=0}^{2N-1} x(n) \exp\{-j\frac{(2k+1)(2n+1)\pi}{4N}\} \sin\frac{(2n+1)\pi}{4N}, \quad k = 0, 1, \ldots, N-1. \tag{31}$$

If we define another transform with basis functions only length N,

$$\begin{aligned} t_{nk} &= \frac{1}{N} \exp\frac{j(2n+1)k\pi}{2N} \\ &= \frac{1}{N}\{DCT_{nk} - j \cdot DST_{nk}\}, \qquad n, k = 0, 1, ..., N-1. \end{aligned} \tag{32}$$

then the CLT can be expressed in the form of [13]

$$\begin{aligned} X_{clt}(k) = \frac{1}{2}(-1)^k \sum_{n=0}^{N-1} x(n)[t_{nk} - t_{n(k+1)}] \\ + \frac{1}{2}(-1)^k \sum_{n=N}^{2N-1} x(n)[t_{nk} + t_{n(k+1)}], \end{aligned} \tag{33}$$

This leads to the CLT architecture as shown in Fig. 6, in which the t_{mn} are generated by using the DCT and DST dual generating circuit as depicted in Fig. 3. The number of multipliers and adders required for these IIR filter structures are summarized in Table 4.

The architecture to generate 1-D DXT is depicted in Fig. 7. This parallel structure consists of a shift register array of size N, two adders, and N IIR filter modules. Two sets of inputs $x(t+N) - x(t)$ and $-x(t+N) - x(t)$ are generated for the even and odd filter modules respectively. When a new datum $x(t)$ arrives, a new set of transform coefficients are obtained in O(1) time, *i.e.* the throughput rate is O(1).

4 Architectures for Inverse Transforms

Inverse transforms are important in retrieveing original information in digital communication systems. The inverse DHT and DFT are given by

$$x_h(k,t) = \frac{1}{\sqrt{N}} \sum_{k=t}^{t+N-1} X_h(k) cas\left(2n\frac{\pi(k-t)}{N}\right)$$

$$k = 0, 1, ..., N-1. \tag{34}$$

$$x_f(k,t) = \frac{1}{\sqrt{N}} \sum_{k=t}^{t+N-1} X_f(k) \exp\{j2n\frac{\pi(k-t)}{N}\}, \quad k = 0, 1, \ldots, N-1. \tag{35}$$

Comparing IDFT, IDHT with their forward transforms, we observe that the transfer function of IDHT is exactly the same as the forward transfrom. The transfer function of the IDFT is

$$H_f(z) = \frac{1}{\sqrt{N}} \left(1 - z^{-N}\right) \left(\frac{\cos\frac{2\pi k}{N} - j\sin\frac{2\pi k}{N} - z^{-1}}{1 - 2\cos\frac{2\pi k}{N} z^{-1} + z^{-2}} \right), \tag{36}$$

which is same as (27) except the imaginary part is negative of original values. Therefore, the IDHT and IDFT can be realized by using the same architecture as the DHT and DFT as depicted in Fig. 4 except adding an invertor at the output of the $ImX_f(k,t)$.

The inverse DCT and DST (IDCT and TDST) are defined as follows:

$$x_c(n,t) = \sqrt{\frac{2}{N}} \sum_{k=t}^{t+N-1} C(k-t) X_c(k) \cos\left[(n+\frac{1}{2})\frac{(k-t)\pi}{N}\right], \tag{37}$$

$$n = 0, 1, ..., N-1. \tag{38}$$

$$x_s(n,t) = \sqrt{\frac{2}{N}} \sum_{k=t+1}^{t+N} C(k-t) X_s(k) \sin\left[(n+\frac{1}{2})\frac{(k-t)\pi}{2N}\right], \tag{39}$$

$$n = 0, 1, ..., N-1. \tag{40}$$

Because $C(k)$ is inside the transform, the architectures require some modification. Since $C(k) = 1$ except for $k = 0$ or $k = N$, we can rewrite (37) as

$$x_c(n,t) = \sqrt{\frac{2}{N}} \sum_{k=t}^{t+N-1} X(k) \cos\left[(n+\frac{1}{2})\frac{(k-t)\pi}{N}\right] + \sqrt{\frac{2}{N}}(\sqrt{\frac{1}{2}} - 1)X(t). \tag{41}$$

We observe that the first part of the above equation is in the form of DCT except the leading constant coefficents $C(k)$. Hence, the transfer function of the IDCT is that of the DCT plus one delay term as follows

$$H_{ic}(z) = \sqrt{\frac{2}{N}} \left((-1)^k - z^{-N}\right) \cos\frac{\pi k}{2N} \frac{\left(1 - z^{-1}\right)}{\left(1 - 2\cos\frac{\pi k}{N} z^{-1} + z^{-2}\right)} + \sqrt{\frac{2}{N}}(\sqrt{\frac{1}{2}} - 1) z^{-(N-1)} \tag{42}$$

This implies that the IDCT can be implemented by using the same architecture as the DCT except adding the compenated term $\sqrt{\frac{2}{N}}(\sqrt{\frac{1}{2}} - 1)X(t)$. The architecture is shown in Fig. 8. Similarly,

the IDST can be rewritten as

$$x_s(n,t) = \sum_{k=t}^{t+N-1} X(k) \sin\left[\frac{\pi(2n+1)(k-t)}{2N}\right] + \sqrt{\frac{2}{N}}(\sqrt{\frac{1}{2}} - 1)X(t+N-1), \tag{43}$$

whose transfer function is

$$H_{is}(z) = \sqrt{\frac{2}{N}}\left((-1)^k - z^{-N}\right) \sin\frac{\pi k}{2N} \frac{(1-z^{-1})}{(1-2\cos\frac{\pi k}{N}z^{-1}+z^{-2})} + \sqrt{\frac{2}{N}}(\sqrt{\frac{1}{2}} - 1) \tag{44}$$

The architecture for IDST is shown in Fig. 9.

The Inverse Complex Lapped Transform (ICLT) [13] of samples $[X(t), X(t+1), ..., X(t+N-1), X(t+N), ..., X(t+2N-1)]$ is defined as

$$x_{clt}(k,t) = \frac{1}{2\sqrt{N}} \sum_{k=t}^{t+N-1} [X(k) + X(k-1) \tag{45}$$
$$+(-1)^k(X(k+N) + X(k+N-1))] \exp\{j\frac{(2n+1)(k-t)\pi}{2N}\}.$$

The architecture of the IDCLT is depicted in Fig. 10. From the previous derivation, we see that the inverse transforms can be obtained by using the same architectures of the forward transforms with one additional branch of multiplier.

5 Theoretical Basis

The basis functions of all the discrete sinusoidal transforms mentioned above corresponds to a set of orthogonal polynomials. One of the important characteristics of orthogonal polynomials is that any three consecutive polynomials are related by the *Fundamental Recurrence Formula* [15] given by

$$P_n(k) = (k - c_n)P_{n-1}(k) - \lambda_n P_{n-2}(k). \tag{46}$$

The discrete transforms discussed in the previous section satisfy a simpler version of the recurrence relation. More precisely, the parameters c_n and λ_n are independent of n and the basis function $P_n(k)$ is periodic in n and k of period N. In these cases, the *Fundamental Recurrence Formula* can be rewritten as

$$P_n(k) = (k - c)P_{n-1}(k) - \lambda P_{n-2}(k), n = 0, 1, .., N-1, k = 0, 1, 2, .., N-1. \tag{47}$$

For different discrete sinusoidal transforms, the corresponding parameters k, c, λ in the *Fundamental Recurrence Formula* are stated in Table 5.

Lemma 1 *For all discrete transforms whose basis functions satisfy the Fundamental Recurrence Formula (47), the z-transform of the basis functions can be expressed as a rational function with a second order denominator that is the characteristic equation of the Fundamental Recurrence Formula.*

Proof: Since any $P_n(k)$ depends only on the previous two terms, the first two polynomial terms, $P_{-1}(k)$ and $P_{-2}(k)$, uniquely specify the entire set of basis functions.

Apply z transform on index n to both sides of (47),

$$\begin{aligned} P(z,k) &= \sum_{n=0}^{N-1} z^{-n} P_n(k) \qquad (48) \\ &= \sum_{n=0}^{N-1} \{(k-c)z^{-n} P_{n-1}(k) - \lambda z^{-n} P_{n-2}(k)\} \\ &= (k-c)[P_{-1}(k) + z^{-1} \sum_{n=0}^{N-1} z^{-n} P_n(k) - z^{-N} P_{N-1}(k)] - \\ &\lambda[P_{-2}(k) + z^{-1} P_{-1}(k) + z^{-2} \sum_{n=0}^{N-1} z^{-n} P_n(k) - z^{-N} P_{N-2}(k) - z^{-(N+1)} P_{N-1}(k)] \\ &= (k-c)z^{-1} P(z,k) - \lambda z^{-2} P(z,k) + [(k-c)P_{-1}(k) - \lambda P_{-2}(k)] - \\ &\lambda z^{-1} P_{-1}(k) - z^{-N}[(k-c)P_{N-1}(k) - \lambda P_{N-2}(k)] + \lambda z^{-(N+1)} P_{N-1}(k) \\ &k = 1, 2, .., N-1. \end{aligned}$$

Factoring out $P(z,k)$, we obtain

$$\begin{aligned} P(z,k) &= \frac{z^{-(N-1)} \lambda P_{N-1}(k) - P_N(k) z^{-(N-2)} - \lambda P_{-1}(k) z^1 + P_0(k) z^2}{\lambda - (k-c)z + z^2} \qquad (49) \\ &= \frac{z^2(P_0(k) - P_N(k) z^{-N}) - \lambda z(P_{-1}(k) - P_{N-1}(k) z^{-N})}{\lambda - (k-c)z + z^2}. \end{aligned}$$

Because of the second-order recurrence relation, the denominators of the z-transform of the basis functions are second-order polynomials in z.

The characteristic equation of the *Fundamental Recurrence Formula* (47) is obtained by solving the homogeneous solutions of the difference equation (47). The homogeneous equation is given by

$$P(z,k) = (k-c)P(z,k)z^{-1} - \lambda P(z,k)z^{-2}. \tag{50}$$

Combining both sides of the equation, we have

$$P(z,k)z^{-2}(\lambda - (k-c)z + z^2) = 0. \tag{51}$$

Since $P(z,k)$ dose not equal to zero, we have that $(\lambda - (k-c)z + z^2)$ equals zero and hence the characteristics equation is $(\lambda - (k-c)z + z^2)$, which is the denominator. □

The transfer function of the discrete transforms (DXT) is derived from (1), which is equivalent to

$$X_X(k,t) = C(k)\sum_{n=0}^{N-1} x(n+t)P_n(k), t = 0,1,2,.... \tag{52}$$

Performing the z-transform on the index t on both sides of the above equation, we have

$$H_X(z) \quad = \quad C(k)z^{-(N-1)}\sum_{n=0}^{N-1} z^n P_n(k) = C(k)z^{-(N-1)}P(z^{-1},k) \tag{53}$$

which is the z-transform of the basis orthogonal polynominals with index z replaced by z^{-1} and multiplied by $C(k)z^{-(N-1)}$. That is, the transfer function of the discrete transform can also be expressed as a rational function with a second order denominator

$$H_x(z) = C(k)\frac{(\lambda P_{N-1}(k) - P_N(k)z^{-1} - \lambda P_{-1}(k)z^{-N} + P_0(k)z^{-(N+1)})}{(\lambda - (k-c)z^{-1} + z^{-2})}. \tag{54}$$

Here we illustrate another way to derive the transfer function of the discrete sinusoidal transforms. Substituting the coefficients listed in Table 5 to (54), we can verify the transfer function derived in Section 3.1.

Lemma 2 *To compute the discrete sinusoidal transforms time recursively, we have to factor out the updating component* $(1 - z^{-N})$ *or* $(1 + z^{-N})$ *in the filter realization. There exists an updating component* $(1+z^{-N})$ *or* $(1-z^{-N})$ *in the nominator of the transfer function of the discrete sinusoidal transform, if and only if the boundary conditions of the basis function satisfy* $P_0 = \pm P_N$ *and* $P_{-1} = \pm P_{N-1}$.

Proof: If the updating vector can be realized by $(1+z^{-N})$ or $(1-z^{-N})$, then the nominator of (50) must contain the factor $(1+z^{-N})$ or $(1-z^{-N})$. That is, the nominator can be expressed as

$$\lambda P_{N-1}(k) - P_N(k)z^{-1} - \lambda P_{-1}(k)z^{-N} + P_0(k)z^{-(N+1)} = (1 \pm z^{-N})(a + bz^{-1}), \tag{55}$$

since it is a $(-N-1)$ degree polynominal. Expand the right side of the above equation, we have

$$\lambda P_{N-1}(k) - P_N(k)z^{-1} - \lambda P_{-1}(k)z^{-N} + P_0(k)z^{-(N+1)} = a + bz^{-1} \pm az^{-N} \pm bz^{-N-1}, \tag{56}$$

it follows that

$$\begin{aligned} a &= \mp\lambda P_{-1}(k) = \lambda P_{N-1}(k) \\ b &= \pm P_0(k) = -P_N(k), \end{aligned} \tag{57}$$

and

$$\begin{aligned} P_0(k) &= \pm P_N(k) \\ P_{-1}(k) &= \mp P_{N-1}(k). \end{aligned} \tag{58}$$

This proves the necessary condition. If $P_0 = \pm P_N$ and $P_{-1} = \pm P_{N-1}$, then the nominator in (54) becomes

$$\begin{aligned} &\lambda P_{N-1}(k) - P_N(k)z^{-1} - \lambda P_{-1}(k)z^{-N} + P_0(k)z^{-(N+1)} \\ &= \mp\lambda P_{-1}(k) \pm P_0(k)z^{-1} - \lambda P_{-1}(k)z^{-N} + P_0(k)z^{-(N+1)} \\ &= (1 \pm z^{-N})(\lambda P_0(k)z^{-1} \mp \lambda P_{-1}(k)), \end{aligned} \tag{59}$$

which means the nominator contains the factor $(1 \pm z^{-N})$. □

Lemma 3 *All the transforms that satisfies Lemma 1 and Lemma 2 can be realized by an updating FIR filter with transfer function* $(1-z^{-N})$ *or* $(1+z^{-N})$*, and an IIR filter with seocnd order denominator and first order nominator whose coefficients are dependent on* $\lambda, (k-c), P_0$ *and* P_{-1}.

Proof: If Lemma 1 and Lemma 2 are satisfied, the transfer function can be expressed as

$$H_x(z) = C(k)\frac{(1 \pm z^{-N})(\lambda P_{N-1} - P_N z^{-1})}{(\lambda - (k-c)z^{-1} + z^{-2})}. \tag{60}$$

Therefore, the transform can be realized by the filter structure as shown in Fig. 2. The coefficients are

$$\begin{aligned} D1 &= (k-c) \\ D2 &= \lambda \\ N1 &= \lambda P_{N-1} \\ N2 &= -P_N. \end{aligned} \tag{61}$$

□ Lemma 3 implies that if a transform can be computed time-recursively, a maximum of four multipliers required to realize the transfrom. Fig. 2 shows a good example of this case.

Lemma 4 *For the discrete sinusoidal transforms, the roots of the characteristic equation belong to the set of the root of* $(1 \pm z^{-N})$.

Proof: Since the discrete sinusoidal transform is FIR in natural, the roots of the denominators should be canncelled by the zeros of the nominator. In general, the roots of the denomintor are complex conjugate poles bacause of $(k-c)^2 - 4\lambda < 0$. Therefore, the poles should be cancelled by the zeros of the $(1 \pm z^{-N})$, and the roots of the denominator

$$z1, z2 = \frac{(k-c) \pm \sqrt{(k-c)^2 - 4\lambda}}{2\lambda} \in \left\{ \begin{array}{ll} \exp \frac{j2\pi n}{N} & n = 0, 1, 2, ..., N-1., z^N = 1' \\ \exp \frac{j\pi(2n+1)}{N} & n = 0, 1, 2, ..., N-1., z^N = -1. \end{array} \right\} \tag{62}$$

□

All the discrete sinusoidal transfroms list in Table 4 satisfies Lemmas 1 through 4. Therefore, these transfroms can be computed time recursively and can be realized by a FIR filter with transfer function $(1 \pm z^{-N})$ and an IIR filter with second order polynominals. These facts support the results obtained in Section 3 and 4.

Lemma 5 *If two transform can be dually generated, then they share the same autoregressive model in their IIR filter structure.*

Proof: The basis polynomial p_n and q_n of the dual generated transform pairs satisfy the following equations

$$p_n = D_{xc} p_{n-1} + D_{xs} q_{n-1} \tag{63}$$

$$q_n = D_{xc}q_{n-1} - D_{xs}p_{n-1}.$$

Since p_n and q_n are dually generated and from (64), they have the same characteristic equation. That is

$$I - Az^{-1} = 0, \tag{64}$$

where

$$A = \begin{bmatrix} D_{xc} & D_{xs} \\ -D_{xs} & D_{xc} \end{bmatrix}$$

As shown in Lemma 1, the roots of the denominators are the roots of the characteristics equation. Since p_n and q_n have the same characteristic equation, they have the same denominator. Hence, both transform have identical poles, and as a result, the same autoregressive filter form. □.

Example 1 *The DCT and DST are dual generated transform pairs and share the same second order denominator.*

As shown in [1], the DCT and DST satisfy

$$\begin{aligned} \cos\left[\frac{\pi(2(n+1)+1)k}{2N}\right] &= \cos\left[\frac{\pi k}{N}\right]\cos\left[\frac{\pi(2n+1)k}{2N}\right] - \sin\left[\frac{\pi k}{N}\right]\sin\left[\frac{\pi(2n+1)k}{2N}\right] \\ \sin\left[\frac{\pi(2(n+1)+1)k}{2N}\right] &= \cos\left[\frac{\pi k}{N}\right]\sin\left[\frac{\pi(2n+1)k}{2N}\right] + \sin\left[\frac{\pi k}{N}\right]\cos\left[\frac{\pi(2n+1)k}{2N}\right], \end{aligned} \tag{65}$$

it follows that

$$\begin{aligned} D_{xc} &= \cos\left[\frac{\pi k}{N}\right], \\ D_{xs} &= -\sin\left[\frac{\pi k}{N}\right]. \end{aligned} \tag{66}$$

From (64), the poles are the root the following equation $1 - 2\cos\left[\frac{\pi k}{N}\right]z^{-1} + z^{-2} = 0$, which is the same as the characteristic equation derived from the Lemma 1. This is why the DCT, DST and DFT, DHT share the same second order autoregressive structure. From Lemma 3, it is noted that a maximum of $4N$ multipliers is required to realize the transform. Due to the fact that $\lambda = 1$ and $P_N = \pm P_{N-1}$ for the case of the DCT and DST, we can see that $2N$ multipliers for the DCT and DST is minimum for this realization. Based on Lemma 5, we can combine the denominator together for the dual generation of DCT and DST. This gives an average $1.5N$ multipliers to realize the DCT or DST. We believe that this is the best we can achieve for real time computation.

6 Unified Time-Recursive Based Multi-dimensional Discrete Sinusoidal Transforms

Multi-dimensional transforms provide powerful tools for multi-dimensional signal processing. Some of the important applications are in the areas of signal reconstruction, speech processing, spectrum analysis, tomography, image processing, and computer vision. Specifically, in multispectral imaging, interframe video imaging, and computer tomography, we have to work with three or (higher) dimensional data. It is difficult to generalize the existing fast 1-D algorithms to 3-D or higher dimensional transforms. However, our time-recursive concept can be easily extended to multi-dimensional transforms resulting in architectures that are simple, modular, and hence suitable for VLSI implementation. Since the 3-D DCT is very useful in processing interframe video imaging data, we first describe the filter architecture for the 3-D DCT, and then generalize it to any multi-dimensional discrete sinusoidal transform.

6.1 Time-Recursive Structures for 3-D DCT

The basic concept of time-recursive computation is to compute the new transform at time $(t+1)$ based on the transform at time t. The operations can be divided into two parts, one consists of computing the difference of the input data between time t and $(t+1)$ and the other cocsists of performing the recursive updating. Looking at the basic architecture of computing 1-D DXT as shown in Fig. 2, the basic structure consists of three components: shift registers, adders, and IIR arrays. The shift register is used to store the input data from $x(t)$ to $x(t+N)$; adders are used to compute the difference between data $x(t)$ and $x(t+N)$ and the IIR arrays are used to perform the computation recursively. We will show that the d-D DXT can be computed by using d blocks consisting of shift registers, adders, and filter arrays, each performing the time-recusive computation along a dimension.

For 1-D time-recursive DXT, the input data window is moved one sample at a time. That is, the input data vector at time t is given by the vector $[x(t), ..., x(t+N-1)]$, and at time $(t+1)$ the input data consists of the vector $[x(t+1), ..., x(t+N)]$. The time-recursive relation

for the 2-D transforms is based on updating the input data row by row [16]. A 2D-DCT for HDTV application based on the lattice structure as considered in [16]. The input data sequences for time-recursive 3-D transforms are as shown in Fig. 11. Here, we assume that input data is updated frame by frame in the third axis n_3, that is, the range of the input data $x(n_1, n_2, n_3)$ is $\{n_1 = 0, \cdots, N-1; n_2 = 0, \cdots, N-1; n_3 = 0, 1, 2, ...\}$. We call the input data frame $x(n_1, n_2, t)$ for a specific index t as the tth frame input data. The 3-D DCT of the tth frame input data is defined as

$$\begin{aligned} X_{c^3}(k_1, k_2, k_3, t) &= C(k_1)C(k_2)C(k_3) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=t}^{t+N-1} x(n_1, n_2, n_3) \cos\left[\frac{\pi(2n_1+1)k_1}{2N}\right] \\ &\quad \cdot \cos\left[\frac{\pi(2n_2+1)k_2}{2N}\right] \cos\left[\frac{\pi[2(n_3-t)+1]k_3}{2N}\right]. \end{aligned} \tag{67}$$

The 3-D DCT of the $(t+1)$ frame input data $\{n_1 = 0, \cdots, N-1; n_2 = 0, \cdots, N-1; n_3 = t+1, ..., t+N.\}$ is

$$\begin{aligned} X_{c^3}(k_1, k_2, k_3, t+1) &= C(k_1)C(k_2)C(k_3) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=t+1}^{t+N} x(n_1, n_2, n_3) \cos\left[\frac{\pi(2n_1+1)k_1}{2N}\right] \\ &\quad \cdot \cos\left[\frac{\pi(2n_2+1)k_2}{2N}\right] \cos\left[\frac{\pi[2(n_3-t-1)+1]k_3}{2N}\right]. \end{aligned} \tag{68}$$

The concept of time-recursive approach is to update the 3-D DCT of the $(t+1)$ frame input data based on $X_{c^3}(k_1, k_2, k_3, t)$. The time-recursive relations between the 3-D DCT $X_{c^3}(k_1, k_2, k_3, t+1)$ of the $(t+1)$th input frame and the 3-D DCT $X_{c^3}(k_1, k_2, k_3, t)$ of the (t)th input frame are

$$\begin{aligned} X_{c^3}(k_1, k_2, k_3, t+1) &= \left[X_{c^3}(k_1, k_2, k_3, t) + X_{c^2}[k_1, k_2, t_\Delta]\frac{2}{N}C(k_3)\cos(\frac{\pi k_3}{2N})\right] \\ &\quad \cdot \cos\left(\frac{\pi k_3}{N}\right) + \left[X_{c^2s}(k_1, k_2, k_3, t) + X_{c^2}[k_1, k_2, t_\Delta]\frac{2}{N}C(k_3)\sin(\frac{\pi k_3}{2N})\right] \sin\left(\frac{\pi k_3}{N}\right). \end{aligned} \tag{69}$$

Here we introduce another 3-D transform $X_{c^2s}(k_1, k_2, k_3, t)$ defined as

$$\begin{aligned} X_{c^2s}(k_1, k_2, k_3, t) &= C(k_1)C(k_2)C(k_3) \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \sum_{n_3=t}^{t+N-1} x(n_1, n_2, n_3) \\ &\quad \cos\left[\frac{\pi(2n_1+1)k_1}{2N}\right] \cdot \cos\left[\frac{\pi(2n_2+1)k_2}{2N}\right] \sin\left[\frac{\pi[2(n_3-t)+1]k_3}{2N}\right]. \end{aligned} \tag{70}$$

A similar relation exists between the updated transfrom $X_{c^2s}(k_1, k_2, k_3, t+1)$ and the previous

transform $X_{c^2s}(k_1, k_2, k_3, t)$, that is

$$X_{c^2s}(k_1, k_2, k_3, t+1) = \left[X_{c^2s}(k_1, k_2, k_3, t) + X_{c^2}[k_1, k_2, t_\Delta]\frac{2}{N}C(k_3)\sin(\frac{\pi k_3}{2N})\right] \tag{71}$$
$$\cdot \cos\left(\frac{\pi k_3}{N}\right) - \left[X_{c^3}(k_1, k_2, k_3, t) + X_{c^2}[k_1, k_2, t_\Delta]\frac{2}{N}C(k_3)\cos(\frac{\pi k_3}{2N})\right]\sin\left(\frac{\pi k_3}{N}\right),$$

where t_Δ in $X_{c^2}[k_1, k_2, t_{Delta}]$ implies that the input data(Here we denote as $\Delta(t+N, t)$) of the 2-D DCT are based on difference of two frames and $\Delta(t+N, t)$ is given by

$$\Delta(t+N, t) = (-1)^{k_3}x(n_1, n_2, t+N) - x(n_1, n_2, t), \tag{72}$$

which is a 2-D data frame obtained from the difference between the $(t+N)$th and the tth input data frames as shown in Fig. 11. This is the first part of the time-recursive compuation. The 2-D DCT $X_{c^2}(k_1, k_2, \Delta(t+N, t))$ can be rewritten as

$$X_{c^2}(k_1, k_2, \Delta(t+N, t)) \tag{73}$$
$$= (-1)^k C(k_1)C(k_2)\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x(n_1, n_2, t+N)\cos\left[\frac{\pi(2n_1+1)k_1}{2N}\right]\cos\left[\frac{\pi(2n_2+1)k_2}{2N}\right]$$
$$-C(k_1)C(k_2)\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1} x(n_1, n_2, t)\cos\left[\frac{\pi(2n_1+1)k_1}{2N}\right]\cos\left[\frac{\pi(2n_2+1)k_2}{2N}\right]. \tag{74}$$

The above equation suggests that the 2-D DCT of each frame can be computed first and store it in a shift register array of size $(N+1) \times N^2$. The difference between the 2-D DCT of the tth frame and $(t+N)$th frame is then computed. Equations (70) and (72) indicate that the 3-D DCT can be generated by feeding the 2-D DCT of the updating vector into a lattice module as shown in Fig. 12. The size of the shift register in the lattice module is N^2 because for a specific k_3 there are N^2 values $(k_1 = 0, .., N-1; k_2 = 0, .., N-1)$ to be updated. A similar updating relation exists for the 2-D DCT and the 1-D DCT [16]. The number of shift registers in the lattice module for 2-D and 1-D DCT are N and 1 respectively. In fact, any d-D DCT can be obtained from the 1-D DCT by repeated application of equations (70) and (72). Therefore, the time-recursive 3-D DCT lattice structure consists of three lattice arrays which are used to produce the 1-D, 2-D and 3-D DCT individually. The 3-D DCT can be implemented using either the lattice or the filter structures as described below.

6.1.1 Lattice 3-D DCT architecture

The architecture of the frame-recursive lattice 3-D DCT is depicted in Fig. 13. It consists of three Lattice Array Blocks (LAB0, LAB1, and LAB2) whose configurations are depicted in Fig. 14. The lattice array LABi consists of a shift register array, two adders, and a lattice array; the shift register array is of size $(N+1) \times N^i$ and is used to store the intermediate values. The function of the adders is to update the effect of the new data and eliminate the effect of the previous data. The structure of the lattice array is shown in Fig. 12. The difference between different lattice arrays is only in the number of delays in the feedback loop. There are N^i delay elements in the ith lattice array.

The operation of this architecture can be viewed as follows. Input data is scanned row by row and frame by frame and sent to the first module LAB0 which generates the 1-D DCT of each row on every input frame. When the last datum of each row is available, the 1-D DCT of each input row vector is obtained. These N 1-D DCT transformed data are loaded in parallel into the second module LAB1 every N clock cycles. The LAB1 module is used to generate the 2-D DCT of each data frame. After N^2 clock cycles, when the last datum of the each frame arrives, the 2-D DCT of each frame is available. These values are loaded in parallel into the LAB2 module to generate the 3-D DCT recursively. The difference between the 2-D DCT of the parity of the $(t+N)$th and tth frame is used as the input to the LA2 module. There are N^2 shift registers in the feedback loop of $LA2$ to store the transformed data of each frame. It takes N^2 cycles to finish updating a new 3-D block and this is the period required to obtain a new 2-D DCT data block. It is easy to verify that the system is fully-pipelined.

In applications where only block multi-dimensional transforms are required, the above architecture can be simplified. Intermediate values stored in the shift registers are not necessary. The purpose of the shift registers required is to store the current data obtained from filter arrays, hence its size is reduced to N^i for Lattice Array Block i. Since the updating is unnecessary, the two adders can be eliminated. The lattice block 3-D DCT structure is shown in Fig. 15.

6.1.2 IIR 3-D DCT architecture

We have seen in Section 3 that the lattice structure can be realized as directly as a digital filter by considering the transfer function of each lattice module. This approach is used to convert the time-recursive lattice 3-D DCT structure into its direct form configuration. The only difference between lattice and IIR 3-D DCT architecture is that the lattice array i is replaced by direct form filter array i. The direct form of the lattice module in Fig. 12 is depicted in Fig. 16. The size of the shift register in direct form realization is the same as that of lattice modules. The configuration of the direct form filter 3-D block DCT is depicted in Fig. 16.

6.2 Time-Recursive Structures for Multi-Dimensional DXT

In this section, we generalize the time-recursive concept to any multi-dimensional DXT and derive the fully-pipelined block structures. Denote by $[x(\vec{n_d}, t)]$ the input data file at time t, and by $[x(\vec{n_d}, t+1)]$ the data file at time $(t+1)$ which is obtained by shifting $[x(\vec{n_d}, t)]$ in a direction of one of the axes of $\vec{n_d}$ by one unit. For simplicity, let us assume that the data file is shifted in the direction of the last axis, n_d. The d-dimensional DXT of the input data file $[x(\vec{n_d}, t)]$ is defined as

$$X_{X^d}(\vec{k_d}, t) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} \cdots \sum_{n_d=t}^{t+N-1} x(\vec{n_d}, t) P_{\vec{n_d}}(\vec{k_d}), \tag{75}$$

Here, we assume that the transform kernel $P_{\vec{n_d}}(\vec{k_d})$ is separable[2]. That is

$$P_{\vec{n_d}}(\vec{k_d}) = P_{n_1}(k_1) P_{n_2}(k_2) \cdots P_{n_d}(k_d). \tag{76}$$

From the analysis in Section 6.1, we see that the updated transform $X_{X^d}(\vec{k_d}, t+1)$ is related to the previous transform $X_{X^d}(\vec{k_d}, t)$ by the following equation [16]:

$$X_{XX}(\vec{k_d}, t+1) = \left\{ X_{X^d}(\vec{k_d}, t) + X_{x^{d-1}}[\vec{k}_{d-1}, \Delta(t+N, t)] D_x(k) \right\} \Gamma_x(k), \tag{77}$$

where $\Delta(t+N, t)$ is the difference between the data files at time t and $(t+N)$, and $D_x(k)$ and $\Gamma_x(k)$ are coefficients that depend only on the transform kernel and index k. The above equation indicates that the d-dimensional DXT can be updated recursively using the previous transformed

[2]This is true for all the discrete sinusoidal transfroms considered in this paper.

data $X_{X^d}(\vec{k_d}, t)$ and the $(d-1)$-D DXT of $\Delta(t+N, t)$. This relation can be used recursively such that any d-D DXT can be generated from the 1-D DXT using d filter blocks as shown in Fig. 17.

As described in the previous section, there are two kinds of time-recursive DXT architectures, the moving-frame d-D DXT and the block d-D DXT. The structure of the basic building block in the moving-frame DXT is shown in Fig. 18. where the filter array can be either the lattice or the filter form. The function of each block is to shift the $(d-1)$-dimensional data into a data bank, then distribute the difference of the first and last frame of the data bank to the second stage DXT array. The dimension of the shift register array is $(N+1) \times N^i$ and the delay in filter array i is N^i. The time required to obtain the $(d-1)$-dimensional DXT is N^{d-1}, which is also the time required to obtain the N^d elements of the transformed data.

In the case of block DXT, the size of the shift register array can be reduced and adders can be eliminated because intermediate transformed data do not have to be stored. The size of the shift register array is N^i. The structure of the LAB is shown in Fig. 19. The lattice array i is reset every N^{i+1} cycles.

6.2.1 Area-Time Complexity Analysis

Our architecture for computing the d-dimensional transform DXT over N^d points consists of d blocks, each block is composed of a shift register array followed by a one-dimensional lattice or IIR structure made up of N DXT modules. The ith shift register array is of size $(N+1) \times N^i b$, where $0 \leq i \leq d-1$ and b is the number of bits used to represent each number. The output is generated in a shift register array of size $N^d b$. Therefore the total number of multipliers and adders used is $O(dN) = O(N)$, and the total amount of memory is $O(N^d b)$. The next lemma states that the area of any chip that computes the d-dimensional DFT transform must be $\Omega(N^d b)$, and hence our design asymptotically optimal in its use of area. The same holds true for the remaining transforms. We are using the standard VLSI model as introduced by Thompson[30].

Lemma 6 *Any VLSI system that computes the d-dimensional DFT on N^d points requires area $A = \Omega(N^d b)$, where b is the number of bits required to represent each input number.*

The proof of the lemma can be derived from a result in [29] in a straight forward way. Hence our design uses the least amount of memory asymptotically. The speed of our VLSI design cannot be improved asymptotically since it processes the input in real time. Hence our design is asymptotically optimal in both speed and area.

7 Conclusion

In this paper, we proposed optimal time-recursive unified architectures for computing the DCT, DST, DHT, DFT, LOT, and CLT using only half as many multipliers as the unified lattice structure described in [1]. In the lattice structure, two transforms are dually generated simultaneously, while this optimal architecture has the flexibility of generating either one transform or both together. The basic configuration of the optimal unified architectures has a second order autoregressive model. It is optimal in the sense that the number of the multipliers used is minimum and both speed and area are asymptotically optimal. We also gave a theoretical justification of this property using the Fundamental Recurrence Formula. We show that to generate the DCT and DST, only $2N-2$ multipliers are necessary, while in the case of dual generation of the DCT and DST, only $1.5N$ multipliers are required for each transform on average. Finally, we generalized the time-recursive concept to multi-dimensional transforms. The resulting architecture is fully-pipelined, modular, and regular. It requires only d 1-D arrays for computing a d-D DXT.

References

[1] K. J. R. Liu, and C. T. Chiu, "Unified Parallel Lattice Structures for Time-Recursive Discrete Cosine/Sine/Hartley Transforms," to appear on IEEE Trans. in Signal Processing, May 1993.

[2] A. Rosenfeld, and A. C. Kak, *Digital Picture Processing*, 2nd edition, Academic Press, 1982.

[3] A. V. Oppenheim and R. W. Schafer, **Discrete-Time Signal Processing**, Prentice Hall, 1989.

[4] P. H. Ang, P. A. Ruetz, and D. Auld, "Video compression makes big gains," *IEEE Spectrum*, pp. 16–19, Oct. 1991.

[5] R. Yip and K. R. Rao, "On the shift property of DCT's and DST's," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-35, No. 3, pp. 404-406, March. 1987.

[6] C. Chakrabarti, and J. JáJá, "Systolic architectures for the computation of the discrete Hartley and the discrete cosine transforms based on prime factor decomposition," IEEE Trans. on Computer, vol. 39, No. 11, pp. 1359-1368, Nov. 1990.

[7] W. Kou and J.W. Mark, "A new look at DCT-type transforms," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-37, No. 12, pp. 1899-1908, Dec. 1989.

[8] N. I. Cho and S. U. Lee, "DCT algorithms for VLSI parallel implementations," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-38, No. 1, . 1899-1908, Dec. 1989.

[9] L. W. Chang and M. C. Wu, "A unified systolic array for discrete cosine and sine transforms," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-39, No. 1, pp. 192-194, Jan. 1991.

[10] Z. Wang, "Fast algorithms for the discrete W transform and for the discrete Fourier transform," IEEE Trans. Acoust., Speech, Signal processing, vol. ASSP-32, Aug. 1984.

[11] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. Comput., vol. C-23, pp. 90-93, Jan. 1974.

[12] R. N. Bracewell, "Discrete Hartley transform," J. opt. Soc. Amer., vol. 73, pp. 1832-1835, Dec. 1983.

[13] R. Young, and N. Kingsbury, "Motion Estimation using Lapped Transforms," IEEE ICASSP Proc., pp. III 261-264, March, 1992.

[14] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Processing*, pp. 553–559, Apr. 1989.

[15] T. S. Chihara, *An Introduction to Orthogonal Polynomials.* Gordon and Breach, 1978.

[16] C. T. Chiu, and K. J. R. Liu, "Real-Time Parallel and Fully-Pipelined Two-Dimensional DCT Lattice Structures with Application to HDTV Systems," *IEEE Trans. on Circuits and Systems for Video Technology*, pp. 25–37, March 1992.

[17] K.J.R. Liu,"Novel parallel architectures for shot-time Fourier transform," SRC TR-92-4, University of Maryland, 1992.

[18] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp 1455-1461, Oct. 1987.

[19] R. N. Bracewell, "The fast Hartley transform," Proc. IEEE, vol. 72, pp1010-1018, Aug. 1984.

[20] W. H. Chen, C. H. Smith, and S. C. Fralick, ' 'A fast computational algorithm for the discrete cosine transform," IEEE Trans. Communication, vol. COM-25, pp. 1004-1009, Sept. 1977.

[21] M. Vetterli and H. Nussbaumer, "Simple FFT and DCT algorithm with reduced number of operations," Signal Processing, vol. 6, no. 4, pp. 267-278, Aug. 1984.

[22] H. W. Jones, D. N. Hein, and S. C. Knauer, "The Karhunen-Loeve, discrete cosine and related transform via the Hadmard transform," in Proc. Inc. Telemeter, Conf., Los Angeles, CA, pp. 87-98, Nov. 1978.

[23] W. H. Chen, C. H. Smith, and S. C. Fralick, ' 'A fast computational algorithm for the discrete cosine transform," IEEE Trans. Communication, vol. COM-25, pp. 1004-1009, Sept. 1977.

[24] B. G. Lee, "A new algorithm to compute the discrete cosine transform," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-32, pp 1243-1245, Dec. 1984.

[25] H. S. Hou, "The fast Hartley transform algorithm," IEEE Trans. on Computer, vol. C-36, No. 2, pp. 147-156, Feb. 1987.

[26] R. W. Owens and J. J'aJ'a, "A VLSI chip for the Winograd/Prime factor algorithm to compute the discrete Fourier transform," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-34, pp. 979-989, Aug. 1986.

[27] H. V. Sorenson, *et. al.*, "On computing the discrete Hartley transform," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-33, No. 4, pp. 1231-1238, Oct. 1985.

[28] S. B. Narayanan and K. M. M. Prabhu, "Fast Hartley transform pruning," IEEE Trans. Acous., Speech, Signal Processing, vol. ASSP-39, No. 1, pp. 230-233, Jan. 1991.

[29] P.Duris et. al, "Tight chip area lower bounds for discrete Fourier and Walsh-Hadamard transformations," Information Processing letters 21, pp. 245-247, 1985.

[30] C. D. Thompson, "A complexity theory for VLSI," PH.D. Thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh, 1980.

	Liu-Chiu1	Liu-Chiu2	chen[23] *et. al.*	Lee[24]	Hou[18]
No. of Multipliers	$6N-4$	$4N$	$Nln(N)$ $-3N/2+4$	$(N/2)\ln(N)$	$N-1$
latency	N	$2N$	$N/2$	$[\ln(N)(\ln(N)-1)]/2$	$3N/2$(order)
limitation on transform size N	no	no	powerof 2	power of 2	power of 2
communication	local	local	global	global	global
I/O operation	*SIPO*	*SISO*	*PIPO*	*PIPO*	*SIPO*

Table 1: Comparison of different DCT algorithms

	Γ_c	Γ_s	D_c	D_s	$e(k)$	$f(k)$
DCT/DST	$\cos(\pi k/N)$	$\sin(\pi k/N)$	$C(k)\sqrt{\frac{2}{N}}$ $\cos(\pi k/2N)$	$C(k)\sqrt{\frac{2}{N}}$ $\sin(\pi k/2N)$	1	1
DHT/DFT	$\cos(2\pi k/N)$	$\sin(2\pi k/N)$	$\sqrt{\frac{1}{N}}$	0	1	1
LOT/CLT	$\cos(\pi/2N)$	$\sin(\pi/2N)$	$\sqrt{\frac{1}{N}}(-1)^k j\cdot$ $\exp -j\theta_k$ $\sin(\pi/4N)$	$\sqrt{\frac{1}{N}}(-1)^k j\cdot$ $\exp -j\theta_k$ $\sin(\pi/4N)$	$\exp j2\theta_k$	$\exp j2\theta_k$

Table 2: Coefficients of the Lattice structure for the DXT

	k	n	$D1$	$D2$	$N1$	$N2$
DCT	k	N	$2\cos(\pi k/N)$	1	$C(k)\sqrt{\frac{2}{N}}$ $\cos(\pi k/2N)$	$-C(k)\sqrt{\frac{2}{N}}$ $\cos(\pi k/2N)$
DST	k	N	$2\cos(\pi k/N)$	1	$-C(k)\sqrt{\frac{2}{N}}$ $\sin(\pi k/2N)$	$-C(k)\sqrt{\frac{2}{N}}$ $\sin(\pi k/2N)$
DHT	0	N	$2\cos(2\pi k/N)$	1	$\sqrt{\frac{1}{N}}[\cos(2\pi k/N)$ $-\sin(2\pi k/N)]$	$\sqrt{\frac{1}{N}}$
DFT	0	N	$2\cos(2\pi k/N)$	1	$\sqrt{\frac{1}{N}}[\cos(2\pi k/N)$ $+j\sin(2\pi k/N)]$	$\sqrt{\frac{1}{N}}$
CLT	0	$2N$	$\exp j2\theta_k$ $2\cos(\pi/2N)$	$\exp j4\theta_k$	$\sin(\pi/4N)^2$ $\exp j\theta_k$	$(-1)^k\sin(\pi/4N)$ $\cos(\pi/4N)\exp j4\theta_k$

Table 3: Coefficients of the universal IIR filter structure for the DXT

Transforms	multipliers	adders
DCT	$2N-2$	$3N+2$
DST	$2N-2$	$3N+2$
DHT	$2N$	$3N+1$
DFT	$3N-2$	$3N+1$
LOT*	$4N$	$4N$
CLT*	$4N$	$4N$
DCT and DST	$3N-3$	$4N+2$
DHT and DFT	$3N-2$	$4N+1$

Table 4: Number of multipliers and adders for different transforms with IIR filter realizations(Here * denotes complex operations).

	k	c	λ	P_0	P_{-1}	P_{N-1}	P_N
DCT	$2\cos(\pi k/N)$	0	1	$\cos(\pi k/2N)$	P_0	$(-1)^k P_0$	P_0
DST	$2\cos(\pi k/N)$	0	1	$\sin(\pi k/2N)$	$-P_0$	$-(-1)^k P_0$	P_0
DHT	$2\cos(2\pi k/N)$	0	1	1	$\cos(2\pi k/2)$ $\sin(2\pi k/N)$	P_{-1}	1
DFT	$2\cos(2\pi k/N)$	0	1	1	$\cos(2\pi k/2)$ $-j\sin(2\pi k/N)$	P_{-1}	1
CLT	$\exp -j2\theta_k$ $2\cos(\pi/2N)$	0	$\exp -j4\theta_k$	1	$\exp j2\theta_k$ $\cos(\pi/2N)$	$(-1)^k$ $\sin(\pi/2N)$ $j\exp j2\theta_k$	$(-1)^k j$

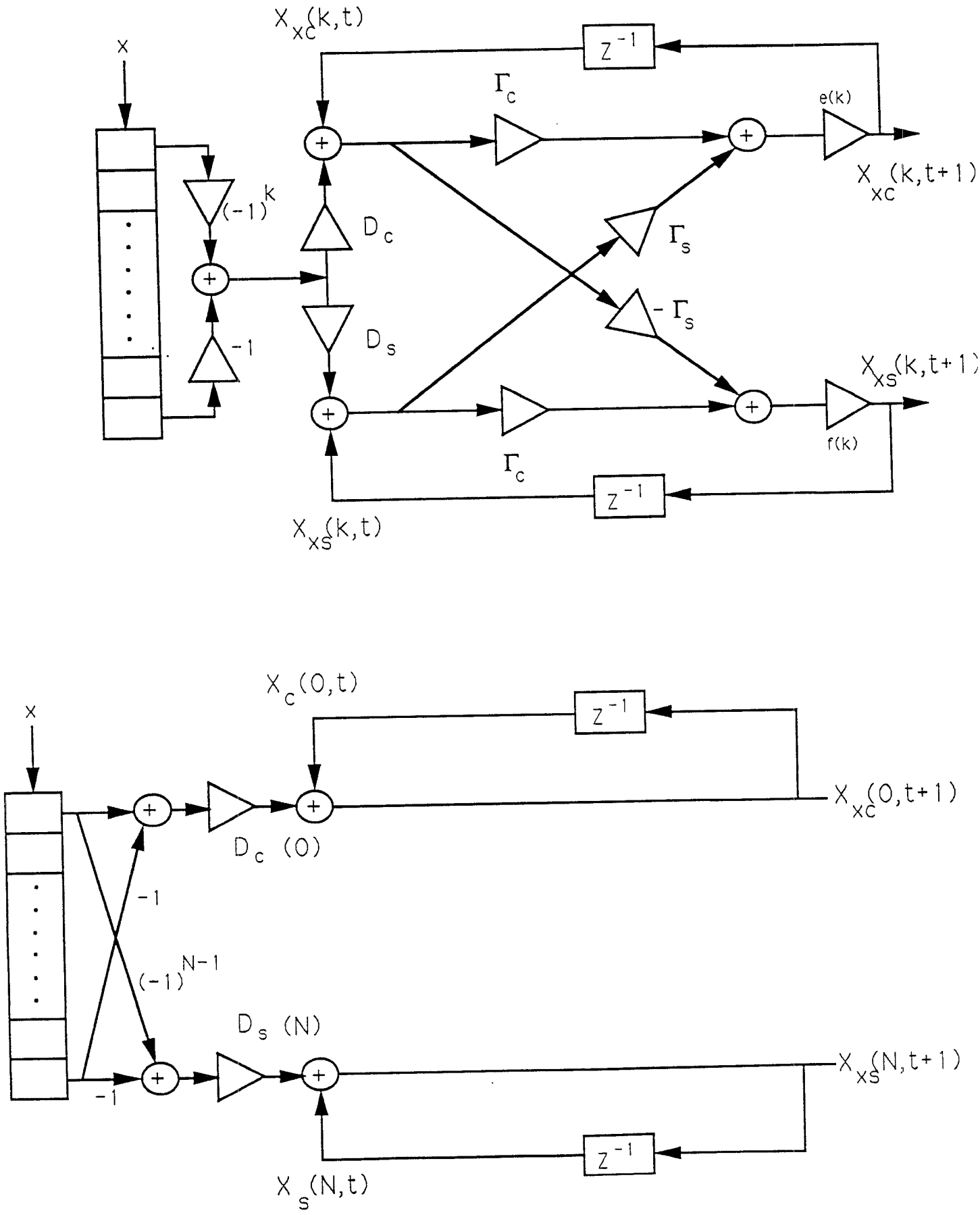


Fig.1 The universal lattice module.

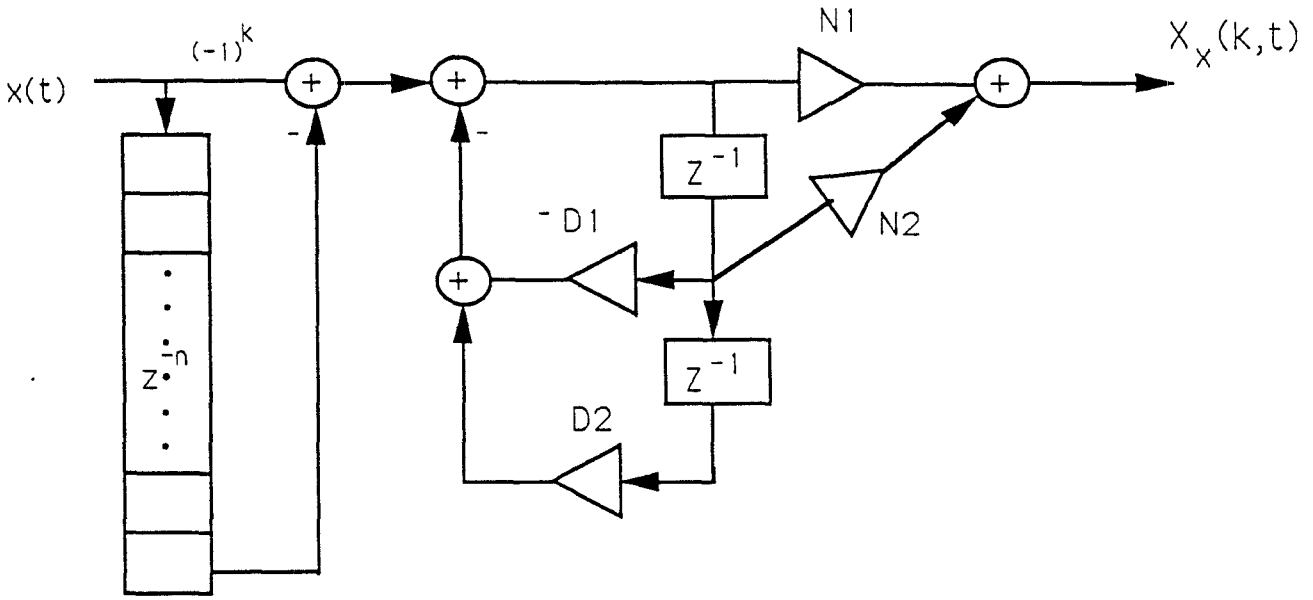


Fig.2 The universal IIR filter module.

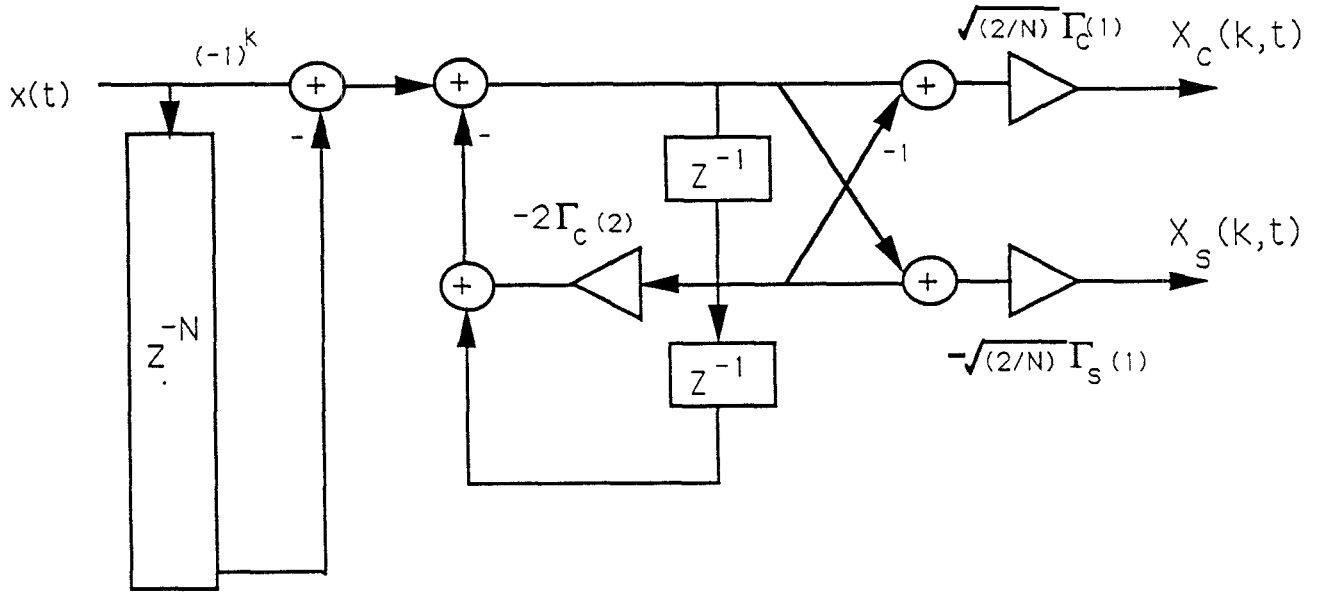


Fig.3 The IIR filter structure for the DCT and DST.

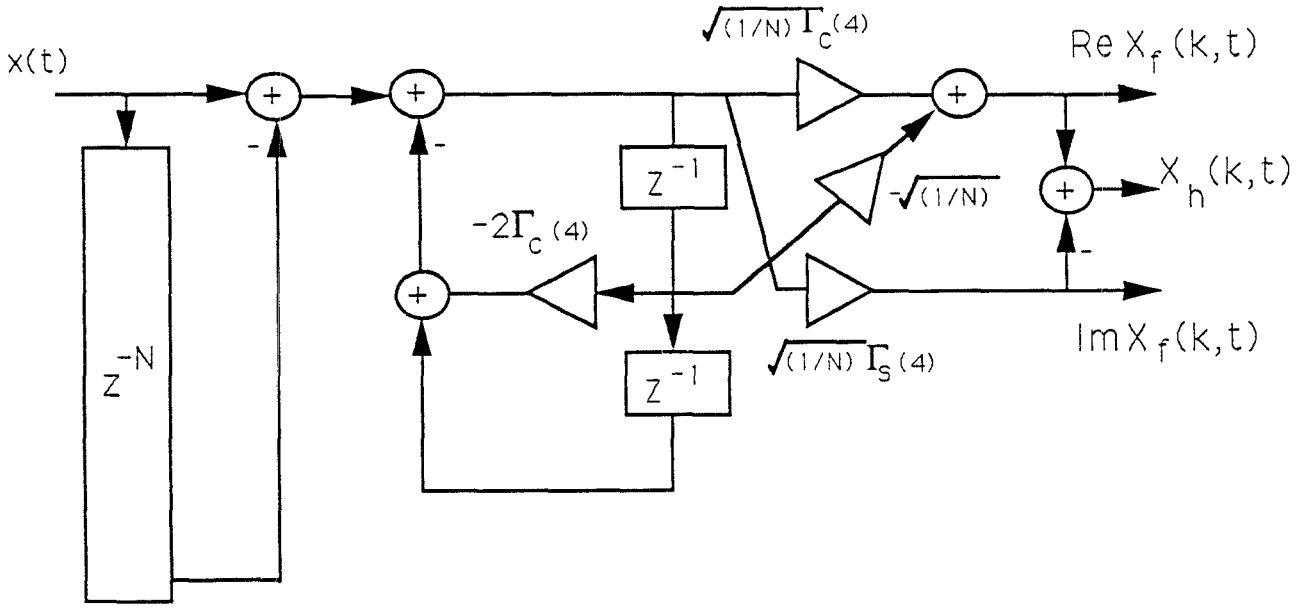


Fig.4 The IIR filter structure for the DHT and DFT.

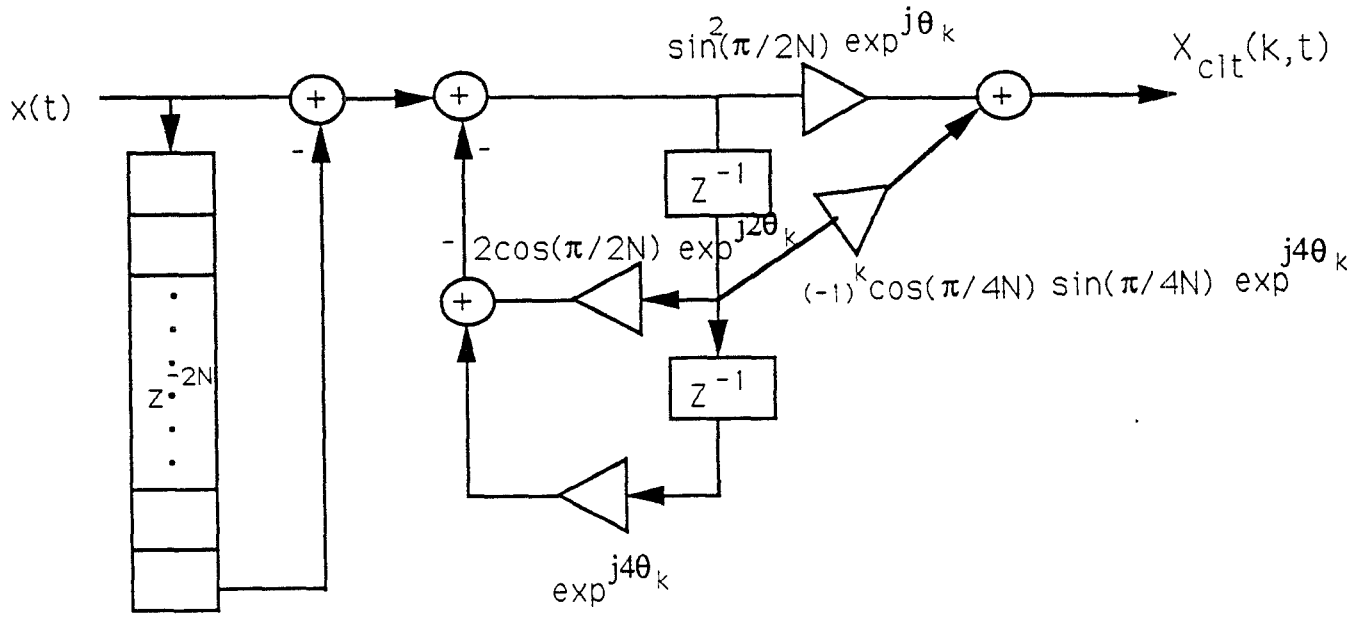


Fig.5 The IIR filter structure for the LOT and CLT.

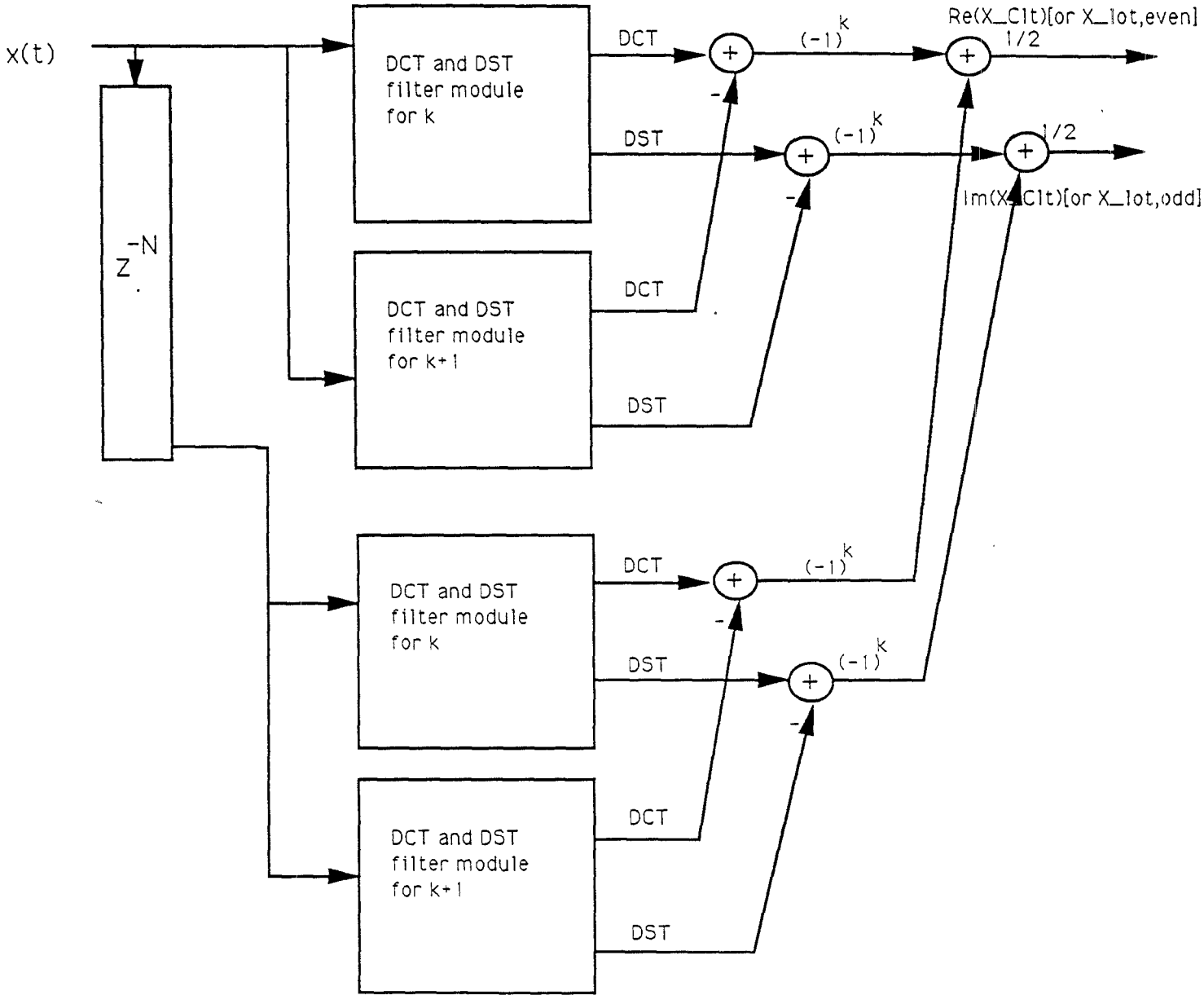


Fig.6 The IIR filter structure for real operation of the LOT and CLT.

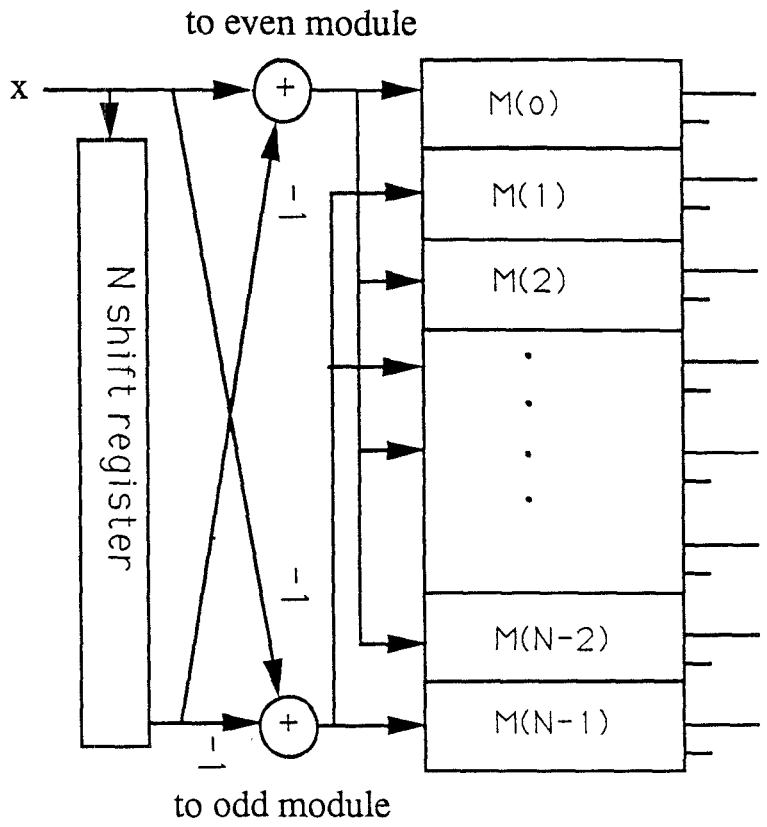


Fig.7 The parallel IIR filter structure for 1-D DXT.

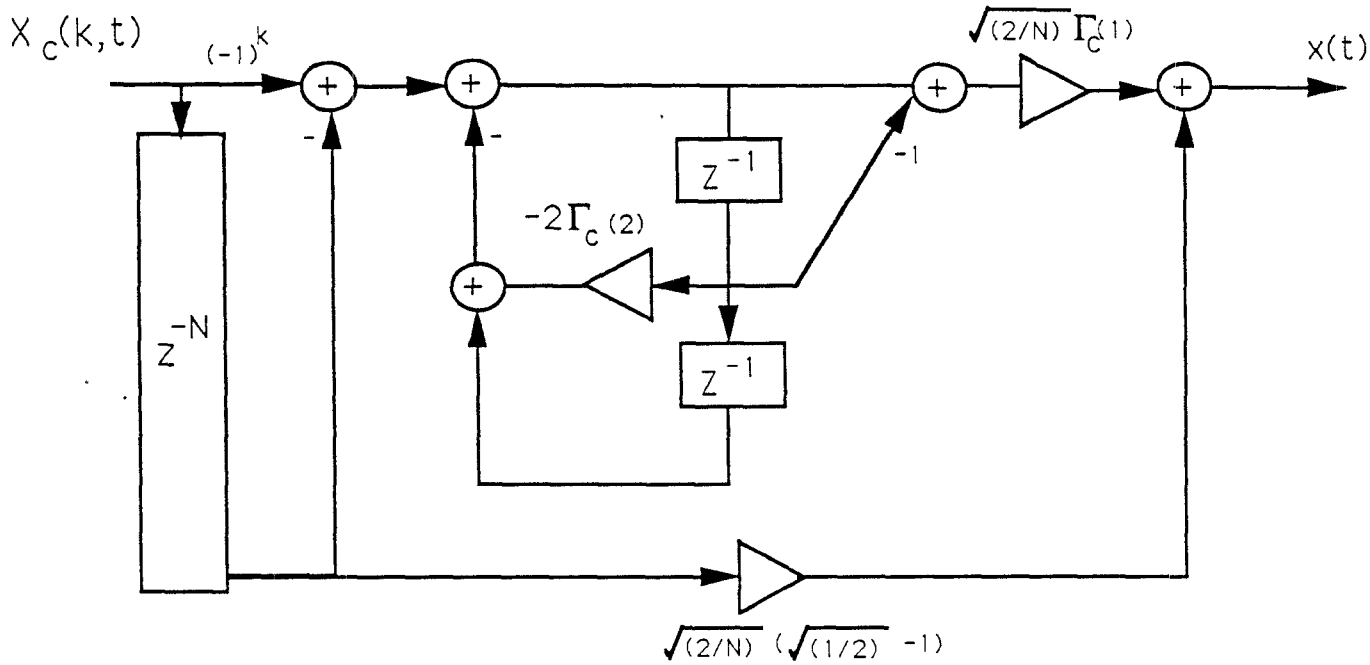


Fig.8 The IIR filter structure for the IDCT.

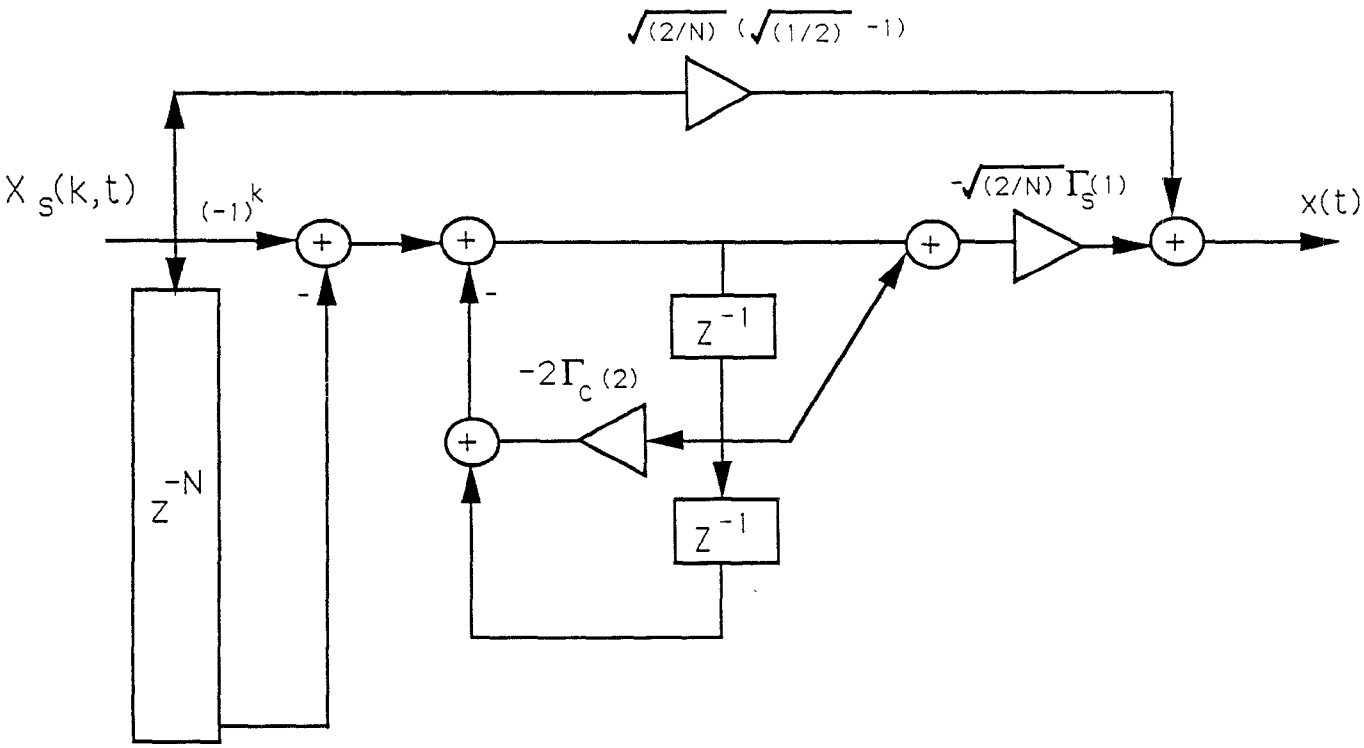


Fig.9 The IIR filter structure for the IDST.

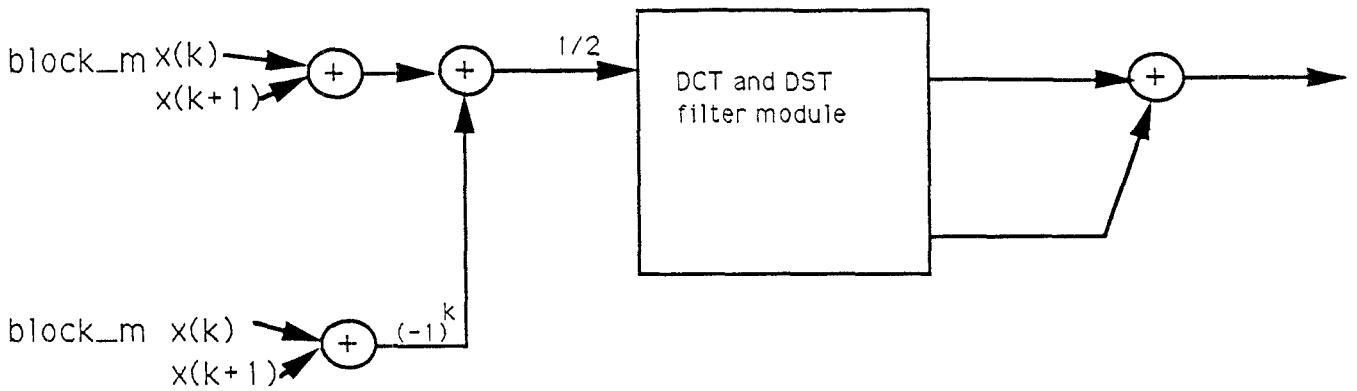


Fig.10 The IIR filter structure for the IDCLT.

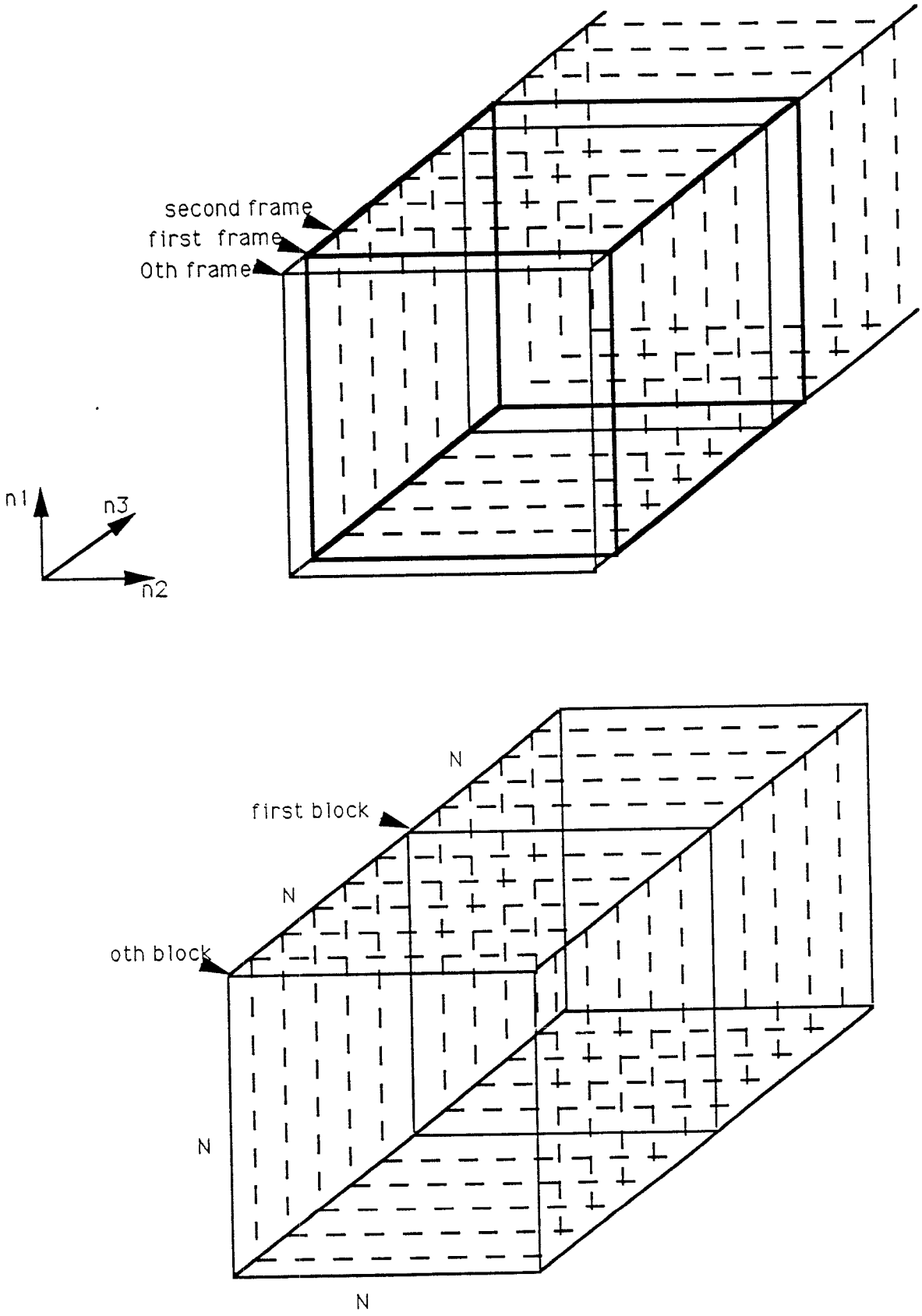


Fig.11 The input sequence of the time-recursive based 3-D transforms.

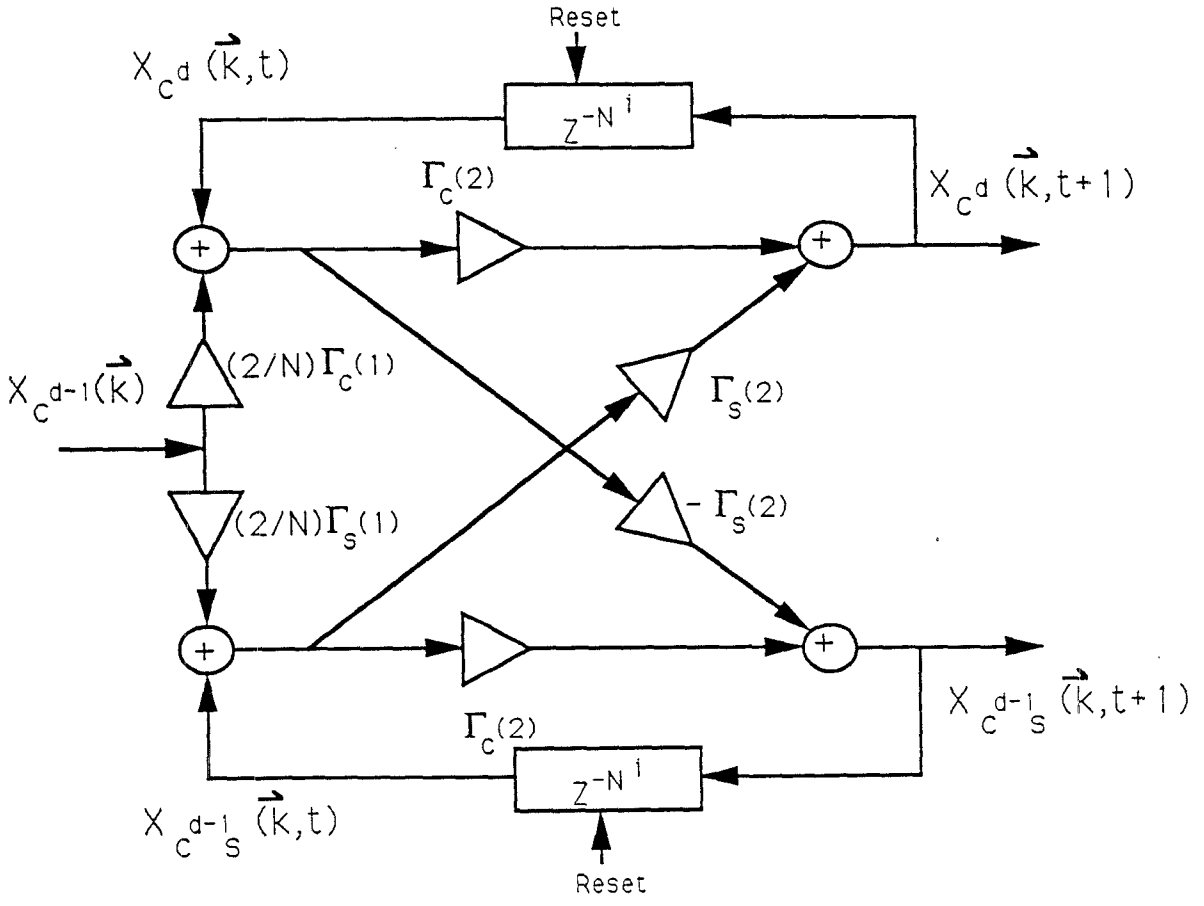


Fig.12 The lattice module.

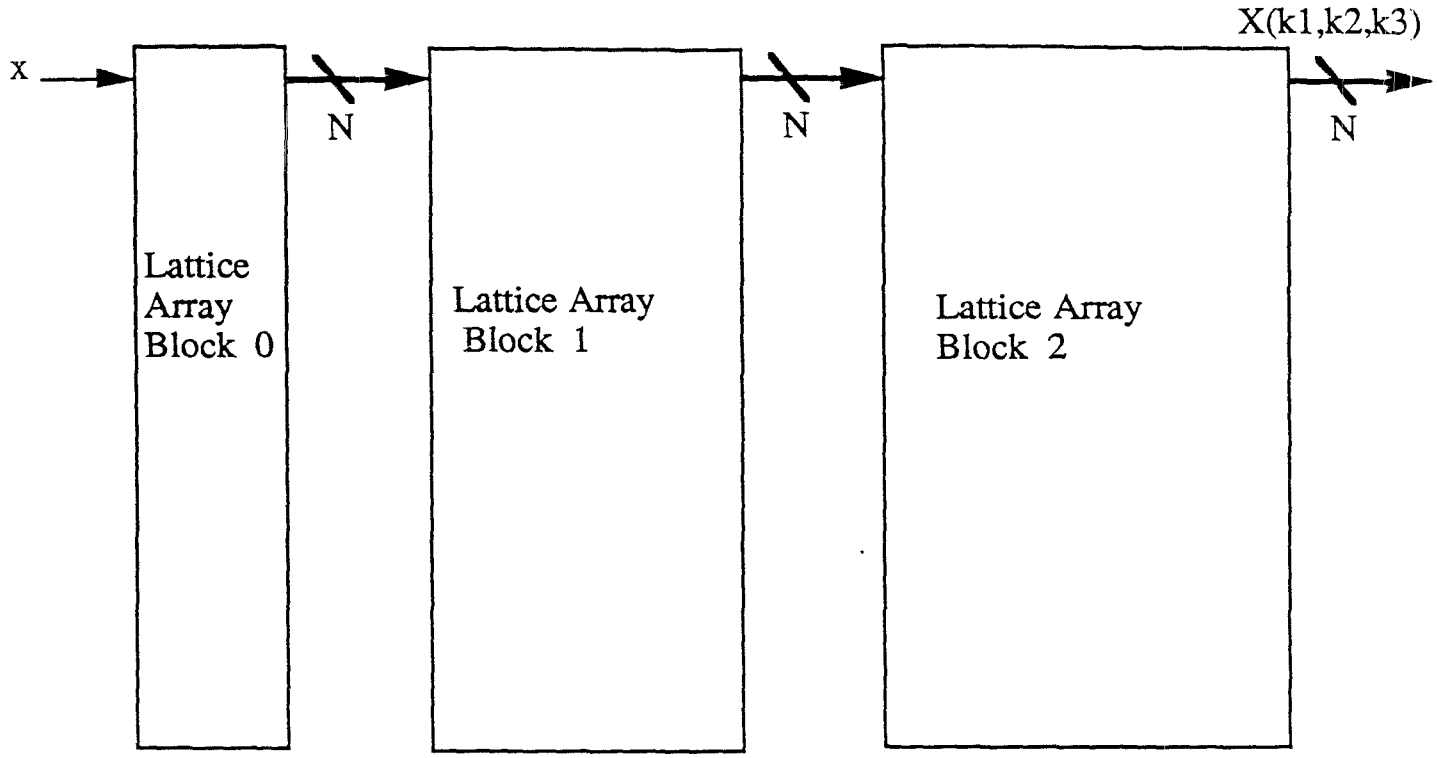
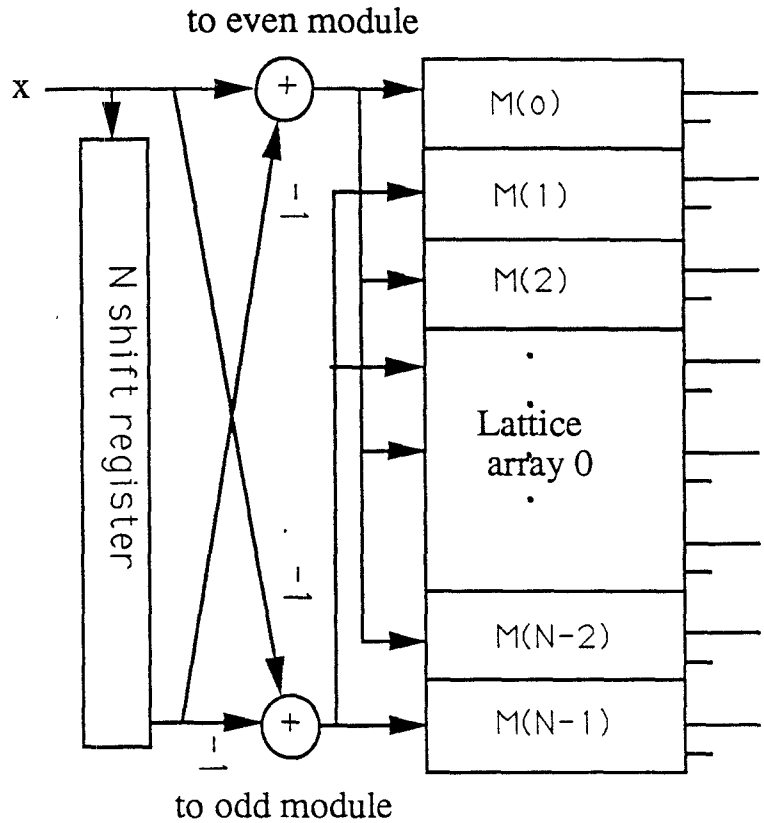
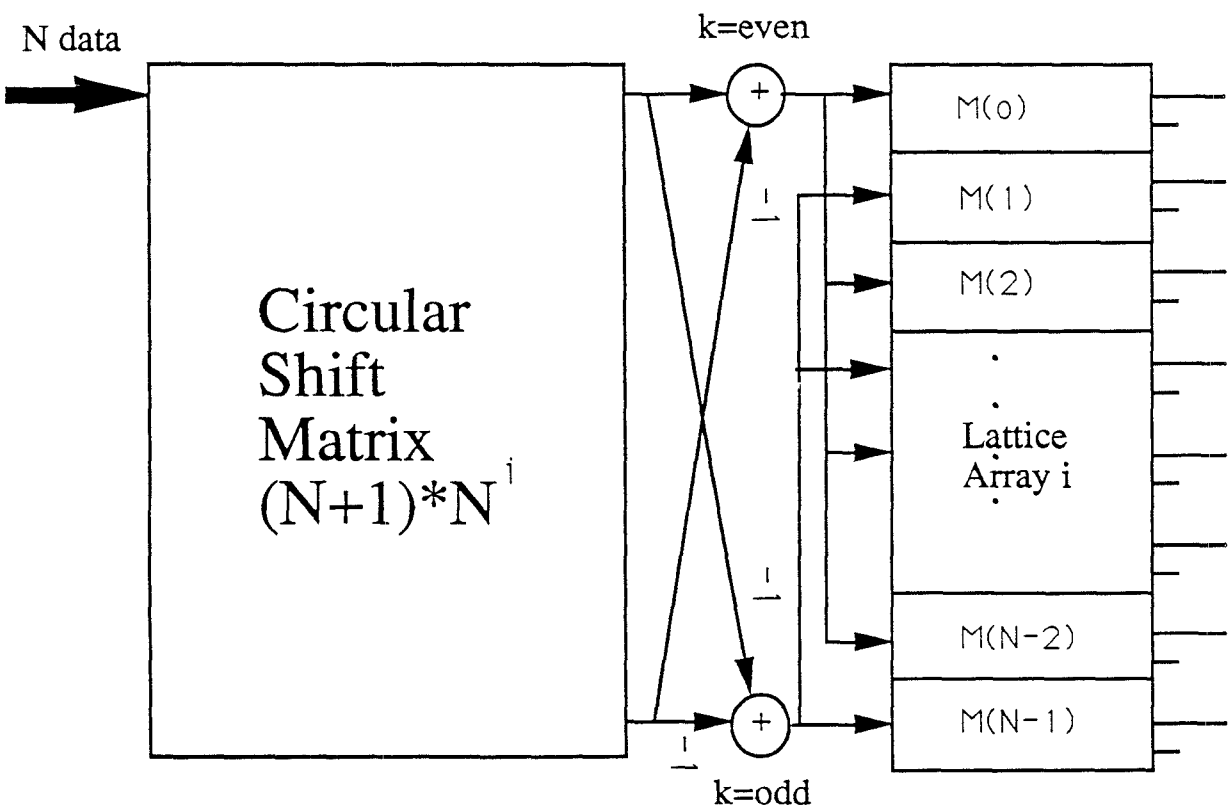


Fig.13 The architecture for the frame-recursive 3-D DCT.



Lattice Array Block 0



Lattice Array Block i, i=1,2,3,...,N-1.

Fig.14 The structure for Lattice Array Blocks.

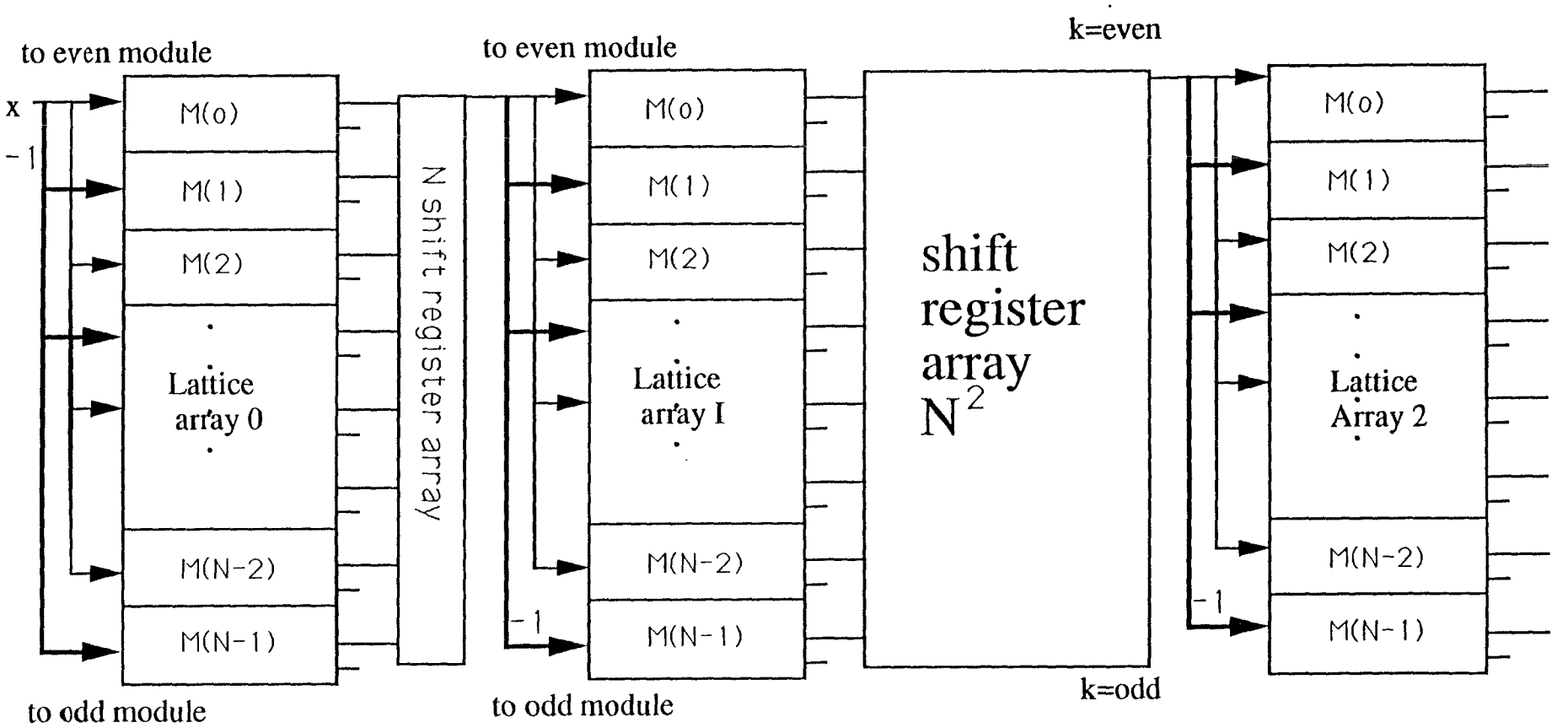


Fig.15 The architecture for block 3-D DCT.

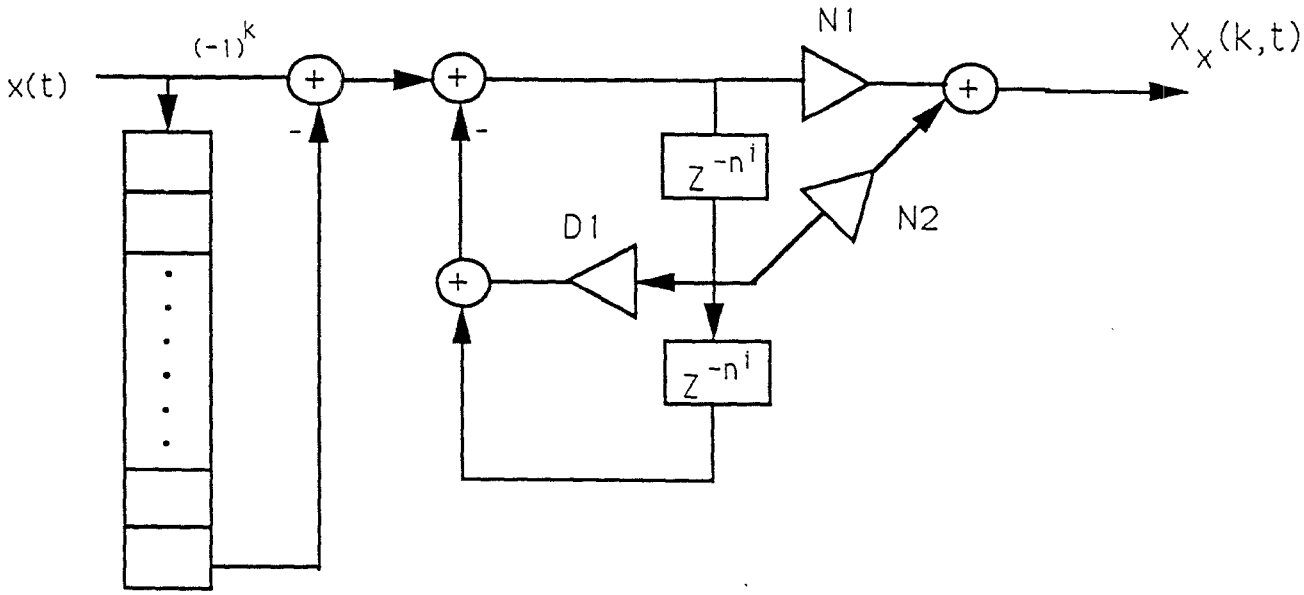


Fig.16 The configuration of the direct form filter 3-D block DCT.

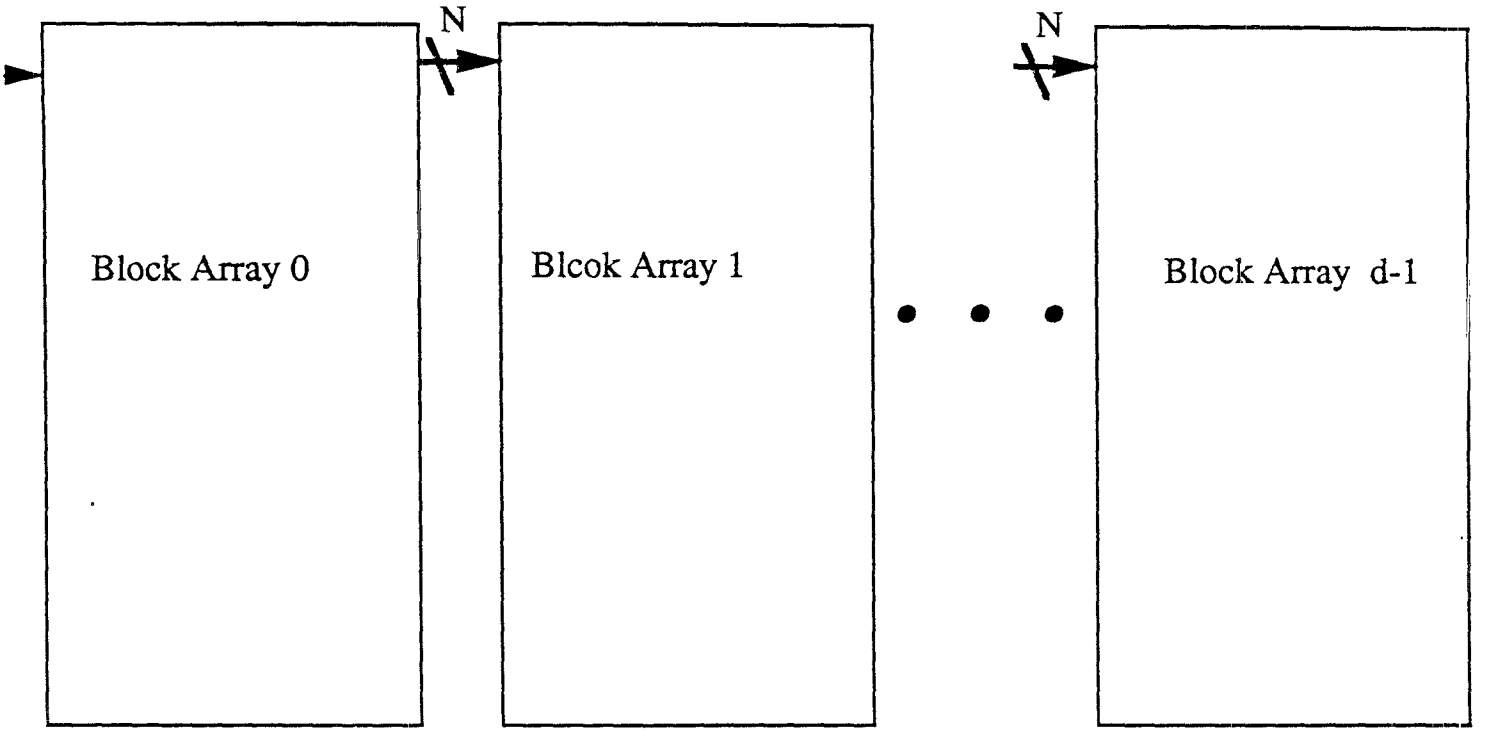
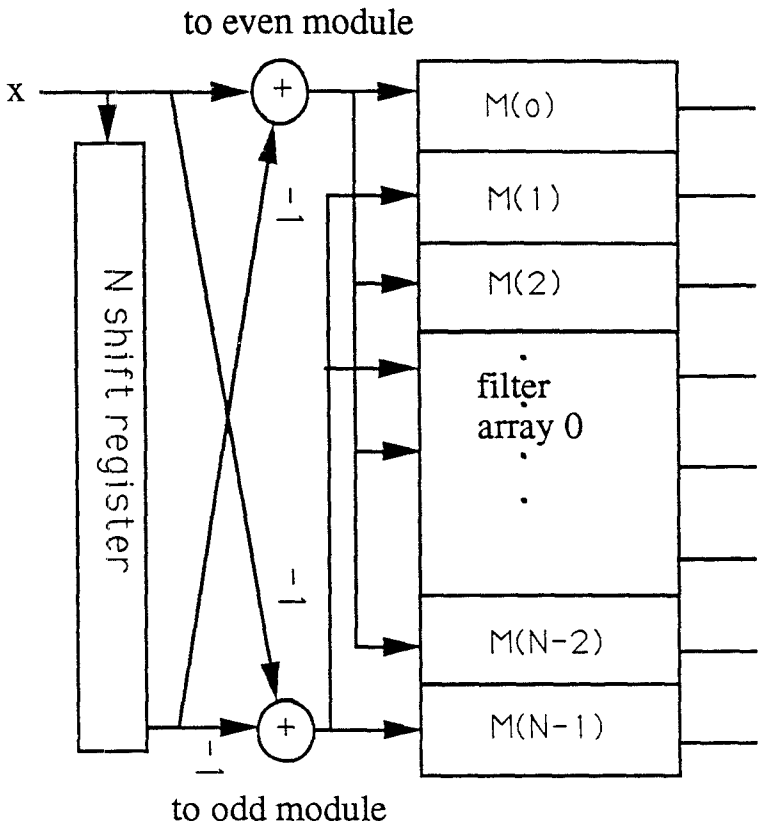
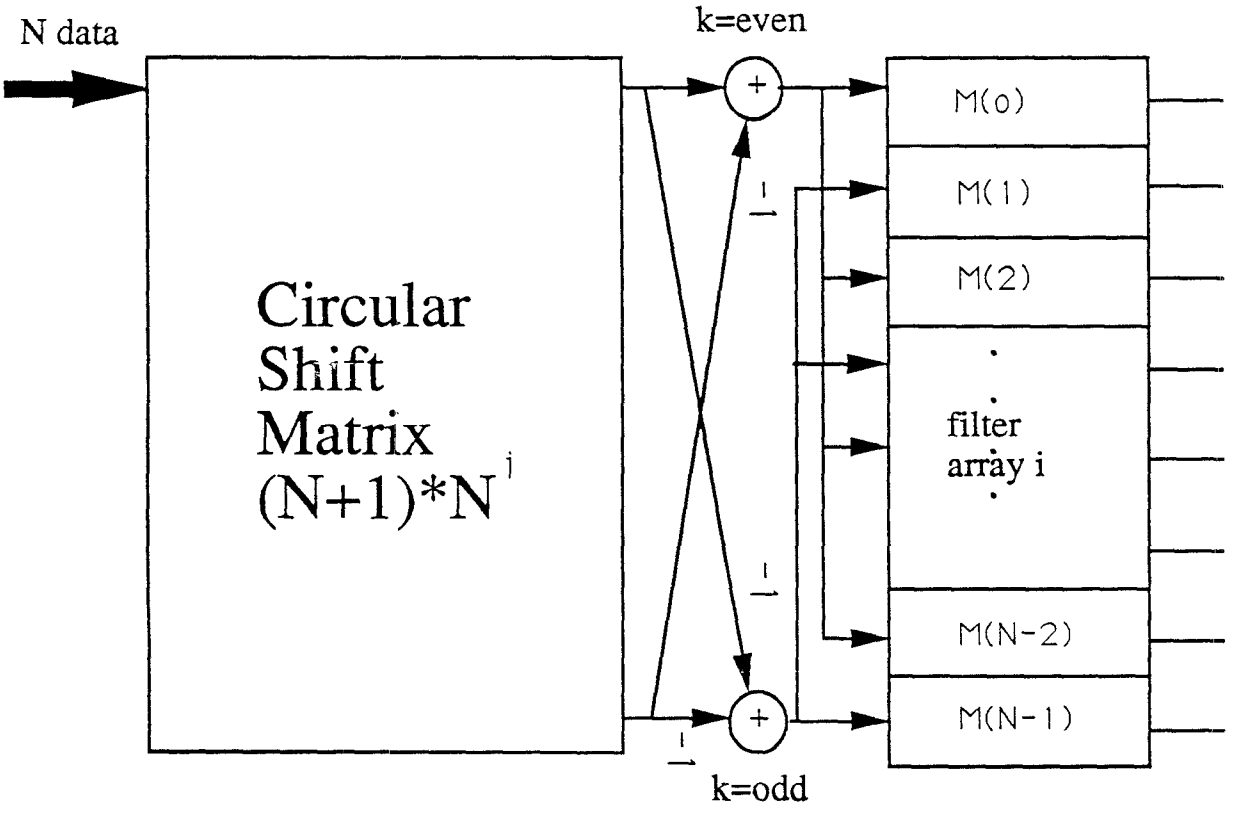


Fig.17 The block diagram of the d-D DXT.

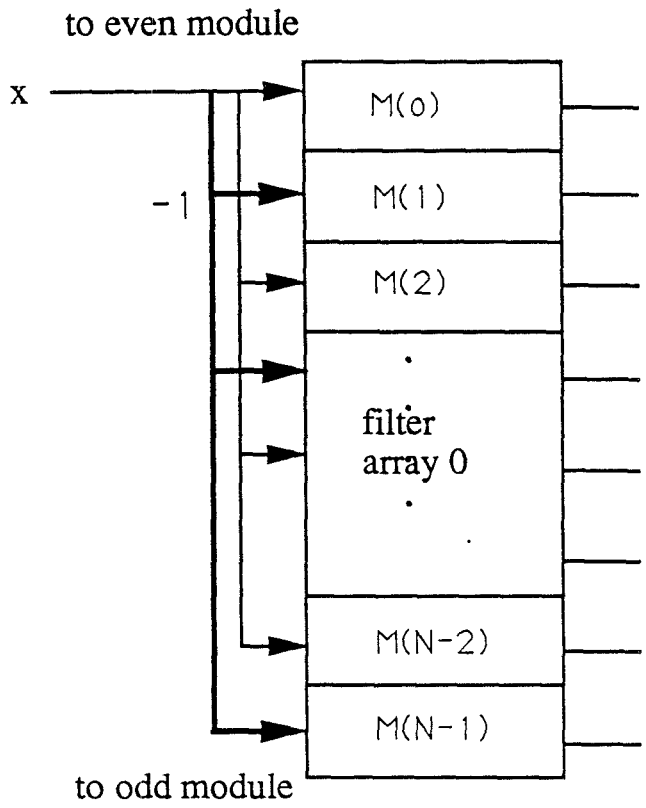


Lattice Array Block 0

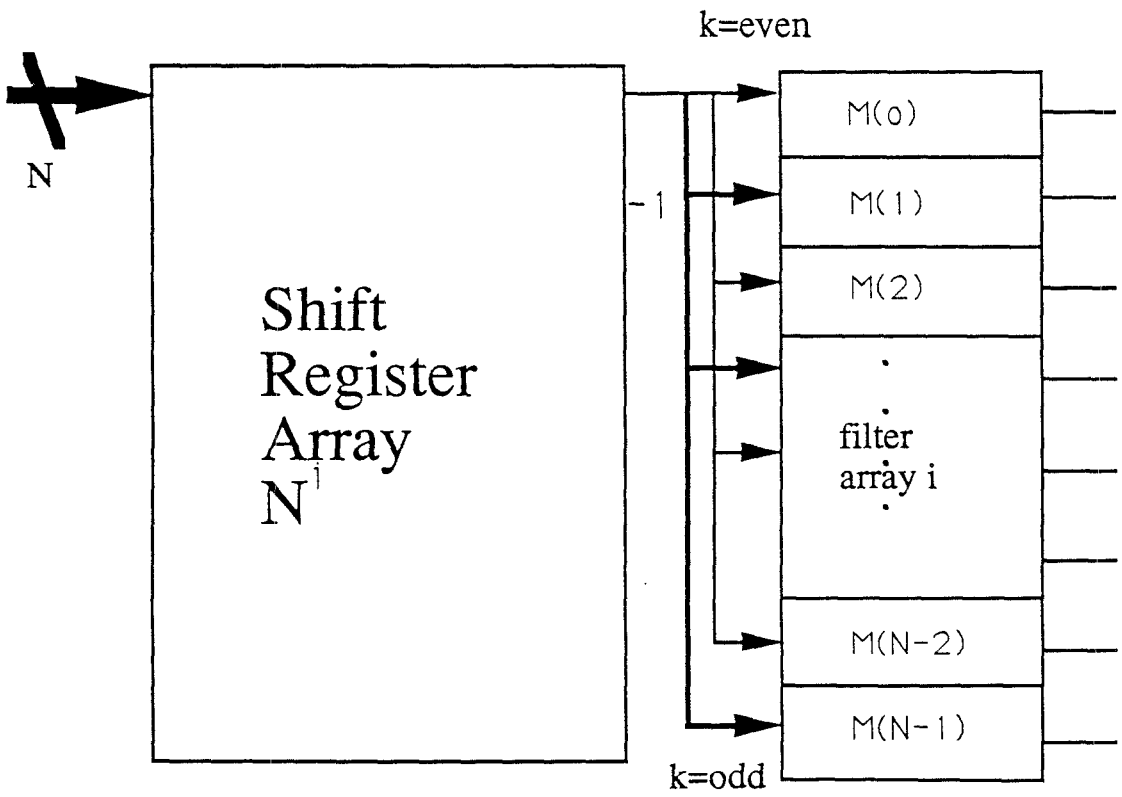


Lattice Array Block i, i=1,2,3,...,N-1.

Fig.18 The basic building structure of the moving-frame DXT.



Filter Array Block 0



Filter Array Block i, i=1,2,3,...,N-1.

Fig.19 The basic structure of the block DXT.