

TECHNICAL RESEARCH REPORT

Algorithm-Based Low-Power Transform Coding Architectures - Part I: The Multirate Approach

by A-Y. Wu and K.J.R. Liu

T.R. 95-32



*Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry*

Algorithm-Based Low-Power Transform Coding Architectures- Part I: The Multirate Approach

An-Yeu Wu and K. J. Ray Liu

Electrical Engineering Department and Institute for Systems Research

University of Maryland

College Park, MD 20742

Phone: (301) 405-6619, Fax: (301) 405-6707

ABSTRACT

In most low-power VLSI designs, the supply voltage is usually reduced to lower the total power consumption. However, the device speed will be degraded as the supply voltage goes down. In this paper, we propose new algorithmic-level techniques for compensating the increased delays based on the multirate approach. We present two methods, the Chebyshev polynomial approach and the polyphase decomposition approach, to design low-power but high-speed transform coding architectures. We will show how to compute the discrete cosine transform (DCT) and its inverse (IDCT) through the decimated low-speed sequences with reasonable linear hardware overhead. For the case the decimation factor equal to two, the overall power consumption can be reduced to about one-third of the original design at the architectural level. Extension of our design to higher decimation rate is also achievable and can result in even lower power consumption. The resulting multirate low-power architectures are regular, modular, and free of global communications. Also, the compensation capability is achieved at the expense of locally increased hardware and data paths. As a consequence, they are very suitable for VLSI implementation. The proposed architectures can also be applied to very high-speed block transforms where only low-speed operators are required. The extensions of the algorithm-based low-power design, such as the unified transform architecture and finite-wordlength effect of the design, will be discussed in the companion paper.

1 Introduction

Recent developments in personal communications services (PCS) have now made it possible to integrate voice, image, and cellular phone networks in a personal communicator. Due to the limited power-supply capability of current battery technology, the power constraint becomes an important consideration in the design of PCS devices. It has been shown that a reduction of the supply voltage is the leveraged decision to lower the power consumption. However, a speed penalty is suffered for the devices (operators) as the supply voltage goes down [1]. In order to meet the low-power/high-throughput constraint, the key issue is to “compensate” the increased delay so that the device can be operated at the slowest possible speed while maintaining the same data sample rate. In [1], the techniques of “parallel processing” and “pipelining” were suggested to compensate the speed penalty, in which a simple comparator circuit was used to demonstrate how parallel independent processing of the data can achieve good compensation at the architectural level. In most digital signal processing (DSP) applications, however, the problems encountered are much more complex. It is almost impossible to directly decompose the problems into independent but parallel tasks. Therefore, the properties of the DSP algorithms should be fully exploited in order to develop efficient compensation techniques to compensate the loss of performance under low-power operations. The main issue here is to reformulate the algorithms so that the desired output can be obtained without hindering the system performance such as data throughput rate. We call such an approach the algorithm-based low-power design. The ultimate goal is to achieve the low-power design requirement only at the expense of larger chip area under current technology, without invoking dedicated arithmetic circuit design, new expensive device material, and advanced VLSI fabrication technology.

In this paper, we will show how to design algorithm-based low-power architectures for transform coding. An algorithm-based compensation technique using the multirate approach is proposed to reduce the power consumption. To motivate the idea, let us consider the discrete cosine transform (DCT) architecture in Fig.1. For most of the existing serial-input-parallel-output (SIPO) DCT algorithms and architectures [2][3], the processing rate must be as fast as the input data rate (Fig.1(a)). In our low-power design, the DCT is computed from the reformulated circuit using the decimated sequences (Fig.1(b)). It is now a multirate system that operates at two different sample rates. Since the operating speed of the processing elements is reduced to half of the original data rate while the data throughput rate is still maintained, the speed penalty is compensated at the architectural level.

As to the power consumption, using the CMOS power dissipation model [1], we can predict that the overall power consumption of the multirate design can be reduced to about one-third of the original system. Therefore, the downsampling scheme provides a direct and efficient way for the low-power design at the algorithmic/architectural level.

Two different approaches to achieve multirate low-power transform kernel design are presented in this paper. One is based on the properties of the Chebyshev polynomial. The Chebyshev polynomial derivation of the DCT/IDCT algorithm was first proposed in [4]. However, the architecture in [4] needs global communication and requires $O(N \log N)$ multipliers. In our work, we treat the transforms as the evaluation of a Chebyshev series. By exploiting the recurrence property of the Chebyshev polynomial, we can compute the DCT/IDCT through the decimated sequences with linear increase of hardware complexity; hence the speed penalty can be compensated. The other is based on the polyphase decomposition approach which is an effective tool in multirate signal processing [5]. By applying the polyphase decomposition to the IIR transfer functions of the DCT/IDCT [3], we can also perform the DCT/IDCT using the multirate approach.

A major advantage of our multirate low-power architectures is that they inherit all advantages of the SIPO transform architectures discussed in [2] and [3] such as local communication, regularity, modularity, and linear hardware complexity. This makes the proposed architectures very suitable for VLSI implementation. Also, the power consumption in the routing and layout of the chip can be minimized. Moreover, unlike most parallel-input-parallel-output (PIPO) transform architectures, the speed-compensation capability of our architectures is achieved at the expense of “locally” increased hardware complexity and routing paths. Since the topological property (communications between chip modules) is usually given the highest priority from the aspects of VLSI system design [6, chap.8], this feature of local interconnection and local hardware overhead is especially preferable when the transformation size is large (*e.g.*, the MPEG audio codec in which a 32-point DCT/IDCT is used [7]).

In the companion paper [8], we will extend the low-power design to a larger class of transforms, which leads to a unified transformation architecture. Some design issues, such as reduction of the complexity for the compensation technique and the finite-wordlength behavior of the low-power design, will also be considered. Those analyses provide us more insights to the algorithm-based low-power design.

The organization of this paper is as follows. The derivation of the low-power IDCT/DCT algo-

gorithms and architectures based on the Chebyshev polynomial is described in Section 2. The multirate IIR DCT/IDCT structures using the polyphase decomposition are presented in Section 3. The comparison of both low-power designs with other approaches is discussed in Section 4 followed by a conclusion.

2 The Chebyshev Polynomial Approach

The n th order Chebyshev polynomial is defined as [9, chap 1]

$$T_n(\eta) = \cos(n\omega), \quad \cos \omega = \eta, \quad -1 \leq \eta \leq 1, \quad (1)$$

which can be generated from the “three-term recurrence” formula

$$T_{n+1}(\eta) = 2\eta T_n(\eta) - T_{n-1}(\eta) \quad (2)$$

with the initial condition $T_0(\eta) = 1$, $T_1(\eta) = \eta$. Now consider the following Chebyshev series

$$Y_c(\eta) = \frac{1}{2}a_0 + \sum_{k=1}^{N-1} a_k \cos(k\omega) = \frac{1}{2}a_0 + \sum_{k=1}^{N-1} a_k T_k(\eta), \quad (3)$$

where a_k , $k = 0, 1, \dots, N-1$, are constant coefficients. One efficient way to evaluate $Y_c(\eta)$ for a given value η is the *Clenshaw's algorithm* [9, chap 3] [10, chap 4], in which a “backward recurrence sequence” is defined as

$$b_k(\eta) = 2\eta b_{k+1}(\eta) - b_{k+2}(\eta) + a_k, \quad \text{for } k = N-1, \dots, 1, 0 \quad (4)$$

with the initial conditions $b_N(\eta) = b_{N+1}(\eta) = 0$. After substituting (4) into (3), and applying the recurrence formula in (2), we can simplify the evaluation of $Y_c(\eta)$ as

$$Y_c(\eta) = \sum_{k=0}^{N-1} [b_k(\eta) - 2\eta b_{k+1}(\eta) + b_{k+2}(\eta)] T_k(\eta) = \frac{b_0(\eta) - b_2(\eta)}{2}. \quad (5)$$

Later in the DCT/IDCT, we will need the evaluation of

$$Y'_c(\eta) = \sum_{k=0}^{N-1} a_k \cos(k\omega) = \sum_{k=0}^{N-1} a_k T_k(\eta). \quad (6)$$

It can be seen that by scaling a_0 by 2 (a left shift) beforehand, we can evaluate $Y'_c(\eta)$ through the same steps in (4)-(5). The corresponding architecture to evaluate $Y'_c(\eta)$ is shown in Fig.2, where a_0 has been pre-scaled by two. Since b_i 's are generated in a "backward" manner, the input sequence is in reverse order. The second-order recurrence structure in the middle computes b_i 's according to (4). After the last input is fed into the system, $b_0(\eta)$ and $b_2(\eta)$ will be available and $Y'_c(\eta)$ can be evaluated from (5) with one addition and one right-shift operation.

Another two Chebyshev polynomial properties that will be useful in later derivations are [9, chap 3]:

1. *Composition property:*

$$T_s(T_r(\eta)) = T_r(T_s(\eta)) = T_{rs}(\eta), \quad (7)$$

which allows us to represent a higher-order Chebyshev polynomial using lower-order ones, and vice versa.

2. *Product-sum relationship:*

$$T_s(\eta)T_r(\eta) = \frac{1}{2}(T_{s+r}(\eta) + T_{s-r}(\eta)), \quad (8)$$

which shows that the product of two Chebyshev polynomials can be decomposed into the sum of two Chebyshev polynomials, and vice versa.

2.1 Chebyshev IDCT Architecture

In order to illustrate the relationship between the Chebyshev polynomial and the transforms, we will begin with the derivation of the IDCT algorithm. Let $X(k)$, $k = 0, 1, \dots, N-1$, be a DCT-domain sequence. The block IDCT to compute the time-domain sequence $x(n)$, $n = 0, 1, \dots, N-1$, is defined as

$$x(n) = \sum_{k=0}^{N-1} C(k)X(k) \cos\left[\frac{(2n+1)\pi}{2N}k\right], \quad (9)$$

where

$$C(k) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } k = 0 \\ \sqrt{\frac{2}{N}}, & \text{otherwise} \end{cases} \quad (10)$$

is the scaling factor used in the DCT/IDCT. If we define

$$\omega_n \triangleq \frac{(2n+1)\pi}{2N} \quad (11)$$

and use the definition of the Chebyshev polynomial in (1), (9) can be written as

$$x(n) = \sum_{k=0}^{N-1} C(k)X(k) \cos(k\omega_n) = \sum_{k=0}^{N-1} X'(k)T_k(\eta_n) \quad (12)$$

where

$$\eta_n \triangleq \cos \omega_n \quad (13)$$

and $X'(k) = C(k)X(k)$ is the scaled input data. Comparing (12) with (6), we see that the IDCT with index n can be treated as the evaluation of Chebyshev series at η_n with coefficients $X'(k)$, $k = 0, 1, \dots, N-1$. As a consequence, the recursive architecture in Fig.2 can perform the IDCT at center frequency ω_n if we replace the multiplier coefficient η with η_n .

Fig.3 shows the IDCT structure based on the Chebyshev evaluation. It has two parts: the *Reverse Array* (RA) and the IDCT module array. The RA consists of one *serial-input-parallel-output* (SIPO) register array and one *parallel-input-serial-output* (PISO) register array. It provides the capability of reversing the input sequence and scaling $X(0)$ in a fully pipelined way. The IDCT module performs (12) at different index n . Since n varies from 0 to $N-1$, we need N IDCT modules to compute the IDCT in parallel. The whole system works in a SIPO way and requires only $N+1$ multipliers and $3N$ adders including the scaling multiplier in RA. The number of multipliers is almost as low as that in Hou's algorithm [11]. Besides, there is no restriction on the block size N and the regularity of our IDCT architecture is more suitable for VLSI implementation.

2.2 Chebyshev DCT Architecture

The DCT of the time-domain block data $x(n)$, $n = 0, 1, \dots, N - 1$, is defined as

$$X(k) = C(k) \sum_{n=0}^{N-1} x(n) \cos\left[(2n+1)\frac{k\pi}{2N}\right], \quad k = 0, 1, \dots, N-1. \quad (14)$$

As with the derivation of the IDCT algorithm, the DCT can be represented as

$$X(k) = C(k) \sum_{n=0}^{N-1} x(n) \cos[(2n+1)\omega_k] = C(k) \sum_{n=0}^{N-1} x(n) T_{2n+1}(\eta_k) \quad (15)$$

where $\omega_k \triangleq \frac{k\pi}{2N}$ and $\eta_k \triangleq \cos \omega_k$. Multiplying $T_1(\eta_k)$ on both sides of (15) and using the Chebyshev property in (8), we obtain

$$\begin{aligned} T_1(\eta_k)X(k) &= C(k) \sum_{n=0}^{N-1} \frac{x(n)}{2} [(T_{2n}(\eta_k) + T_{2n+2}(\eta_k))] \\ &= \frac{C(k)}{2} \sum_{n=0}^N x'(n) T_{2n}(\eta_k) \end{aligned} \quad (16)$$

where

$$x'(n) \triangleq x(n-1) + x(n), \quad n = 0, 1, \dots, N \quad (17)$$

with the assumption of $x(-1) = x(N) = 0$. Recall that $T_1(\eta_k) = \eta_k$ and $T_{2n}(\eta_k) = T_n(T_2(\eta_k))$ (from (7)). If we define

$$\eta'_k \triangleq T_2(\eta_k) = \cos(2\omega_k) = 2\eta_k^2 - 1, \quad (18)$$

$X(k)$ in (16) can be computed as

$$X(k) = \frac{C(k)}{2\eta_k} \sum_{n=0}^N x'(n) T_n(\eta'_k), \quad k = 0, 1, \dots, N-1. \quad (19)$$

Therefore, the DCT at center frequency ω_k can be obtained by evaluating the Chebyshev series at the value η'_k with coefficients $x'(n)$, $n = 0, 1, \dots, N$, followed by the scaling of $\frac{C(k)}{2\eta_k}$.

Note that the DCT of the reversed sequence $\tilde{x}(n) = x(N-1-n)$, $n = 0, 1, \dots, N-1$, is

$$\tilde{X}(k) = C(k) \sum_{n=0}^{N-1} \tilde{x}(n) \cos[(2n+1)\omega_k] = C(k) \sum_{n=0}^{N-1} x(n) \cos\left[k\pi - \frac{(2n+1)k\pi}{2N}\right], \quad (20)$$

for $k = 0, 1, \dots, N - 1$. We can relate $\tilde{X}(k)$ to $X(k)$ by $\tilde{X}(k) = (-1)^k X(k)$. As a result, the RA, which is used to reverse the input sequence, can be eliminated by complementing the odd-indexed $X(k)$'s while keeping the even-indexed $X(k)$'s unchanged. Fig.4 shows the architecture to implement the Chebyshev DCT algorithm. The overall Chebyshev DCT architecture needs a total of $2N - 2$ multipliers and $3N - 1$ adders. It should be noted that the total number of $x'(n)$ is $N + 1$. Therefore, an extra zero is appended after $x(N - 1)$ for the generation of $x'(n)$, $n = 0, 1, \dots, N$. After $x'(N)$ is sent to the DCT array, we can obtain the DCT coefficients in parallel at the array outputs.

2.3 Low-Power Design for the DCT/IDCT

Consider the Chebyshev series in (6) and split it into the even and odd series:

$$\begin{aligned} Y_c'(\eta) &= \sum_{i=0}^{N/2-1} a_{2i} T_{2i}(\eta) + \sum_{i=0}^{N/2-1} a_{2i+1} T_{2i+1}(\eta) \\ &= Y_e(\eta) + Y_o(\eta) \end{aligned} \quad (21)$$

where $Y_e(\eta)$ and $Y_o(\eta)$ denote the even and odd series, respectively. By the use of (2) and (7), $Y_e(\eta)$ can be written as

$$Y_e(\eta) = \sum_{i=0}^{N/2-1} a_{2i} T_i(T_2(\eta)) = \sum_{i=0}^{N/2-1} a_{2i} T_i(\eta') \quad (22)$$

with $\eta' = 2\eta^2 - 1$. On the other hand, $Y_o(\eta)$ can be converted into an even series by following the derivations in (15)-(19):

$$Y_o(\eta) = \sum_{i=0}^{N/2} \kappa(a_{2i-1} + a_{2i+1}) T_i(\eta'). \quad (23)$$

where $\kappa = \frac{1}{2\eta}$ is a pre-calculated constant coefficient. Now combining (22) and (23) together, we have

$$Y_c'(\eta) = \sum_{i=0}^{N/2} [a_{2i} + \kappa(a_{2i-1} + a_{2i+1})] T_i(\eta') = \sum_{i=0}^{N/2} d_i T_i(\eta') \quad (24)$$

with

$$d_i \triangleq \underbrace{a_{2i}}_{\text{even}} + \underbrace{\kappa(a_{2i-1} + a_{2i+1})}_{\text{odd}}, \quad i = 0, 1, \dots, \frac{N}{2}. \quad (25)$$

From (24) we can see that the evaluation of a N -point Chebyshev series can be reduced to a $(N/2+1)$ -point evaluation using the new sequence d_i 's which are composed of decimated sequences. This new evaluation method can be easily applied to the computation of the IDCT/DCT as described in Section

2.1 and Section 2.2. The resulting IDCT architecture is depicted in Fig.5, where $\kappa_n = 1/(2\eta_n)$ and $\eta'_n = 2\eta_n^2 - 1$ with η_n defined in (13). Firstly, a_i 's in (24) are replaced with $X(i)$'s in (9), for $i = 0, 1, \dots, N - 1$, then we use one decimation circuit and one adder to compute the even and odd sequences in (25) from the $X(i)$ sequence in a fully-pipelined way (see the left-hand side of Fig.5). After these two decimated sequences are reversed by the RA, they are combined together to generate d_i 's in (25), and d_i 's are sent to the IDCT module to perform the Chebyshev evaluation in (24). Once the evaluation is completed, we have the IDCT coefficient with index n at the module output. Since the operating frequency halves after the decimator, now we can use two times slower multipliers and adders in this IDCT module with some hardware overhead. Meanwhile, the throughput rate is still retained. Similarly, the multirate Chebyshev DCT architecture can be derived as shown in Fig.6, where $\kappa'_k = 1/(2\eta'_k)$, $\eta''_k = 2\eta'^2_k - 1$, and η'_k is defined in (18).

To achieve downsampling by four, we can recursively compute another new $(N/4 + 1)$ -sequence e_i from d_i , which results in

$$e_i = \kappa' \kappa [(a_{4i-3} + a_{4i+1}) + (a_{4i-1} + a_{4i+3})] + \kappa' (a_{4i-2} + a_{4i+2}) + \kappa (a_{4i-1} + a_{4i+1}) + a_{4i}, \quad (26)$$

for $i = 0, 1, \dots, \frac{N}{4}$,

where $\kappa' = \frac{1}{2\eta'}$ is also a pre-computed constant. One possible realization of (26) is depicted in Fig.7. Once the e_i 's are computed from the decimated sequences a_{4i+k} , $k = 0, 1, 2, 3$, the evaluation of $Y'_c(\eta)$ can be computed as

$$Y'_c(\eta) = \sum_{i=0}^{N/4} e_i T_i(\eta'') \quad (27)$$

with $\eta'' = 2\eta'^2 - 1$. Likewise, based on (26) and (27), we can also construct the multirate IDCT and DCT architectures as shown in Fig.8 and Fig.9, in which only four times slower operators are required to compute the transform coefficients.

2.3.1 Power Estimation for the Low-Power Design

Now let us consider the power dissipation of the low-power architectures. The power dissipation in a well-designed digital CMOS circuit can be modeled as [12]

$$P \approx C_{eff} \cdot V_{dd}^2 \cdot f_{clk}, \quad (28)$$

where C_{eff} is the effective loading capacity, V_{dd} is the supply voltage, and f_{clk} is the operating frequency. Also, the lowest possible supply voltage V'_{dd} can be approximated by [1][13]

$$\frac{V'_{dd}}{(V'_{dd} - V_t)^2} = M \frac{V_{dd}}{(V_{dd} - V_t)^2}, \quad (29)$$

where M is the decimation factor and V_t is the threshold voltage of the device.

Assume that $V_{dd} = 5V$, $V_t = 0.7V$ in the original system. For the 16-point Chebyshev IDCT under normal operation, it requires 18 multipliers and 48 adders. For the low-power 16-point IDCT with $M = 2$, 34 multipliers and 65 adders are required. From (29), it can be shown that V'_{dd} can be as low as 3.1V for the case $M = 2$. Provided that the capacitance due to the multipliers is dominant in the circuit and is roughly proportional to the number of multipliers, the power consumption of the low-power design can be estimated as

$$\left(\frac{34}{18}C_{eff}\right)\left(\frac{3.1V}{5V}\right)^2\left(\frac{1}{2}f\right) \approx 0.36P_0, \quad (30)$$

where P_0 denotes the power consumption of the original system. Similarly, for the case $M = 4$, the 16-point IDCT needs a total of 66 multipliers and 100 adders. Since the lowest possible supply voltage can be 2.1V (from (29)), the total power can be reduced to

$$\left(\frac{66}{18}C_{eff}\right)\left(\frac{2.1V}{5V}\right)^2\left(\frac{1}{4}f\right) \approx 0.16P_0. \quad (31)$$

Therefore, we can achieve low-power consumption at the expense of reasonable complexity overhead. Such a tradeoff will be considered in Section 4.

3 The Polyphase Decomposition Approach

Performing orthogonal transforms based on the IIR transfer function approach was studied in [3]. By considering the transform operator as a linear shift invariant (LSI) system that maps the serial input data into their transform coefficients, the authors in [3] have shown that most discrete sinusoidal transforms can be realized by using a unified IIR structure. In this section, we will show that, in addition to the Chebyshev approach, we can also derive the multirate low-power DCT/IDCT algorithms/architectures by applying the polyphase decomposition [5] to the IIR transfer functions

in [3]. We will see later that the polyphase decomposition approach provides a systematic way for architectural low-power design.

3.1 The IIR DCT Algorithm

The one-dimensional (1-D) DCT of a series of input data starting from $x(t - N + 1)$ and ending at $x(t)$ is defined as

$$X_{DCT,k}(t) = C(k) \sum_{n=0}^{N-1} \cos[(2n+1)\frac{k\pi}{2N}] x(t+n-N+1), \quad (32)$$

for $k = 0, 1, 2, \dots, N-1$. A second-order IIR transfer function can be derived from (32) as [3]

$$H_{DCT,k}(z) = \frac{X_{DCT,k}(z)}{X(z)} = ((-1)^k - z^{-N}) \frac{C(k) \cos \omega_k (1 - z^{-1})}{1 - 2 \cos 2\omega_k z^{-1} + z^{-2}} \quad (33)$$

where $\omega_k \triangleq \frac{k\pi}{2N}$, $X_{DCT,k}(z)$ and $X(z)$ denote the z -transforms of $X_{DCT,k}(t)$ and $x(t)$, respectively. For block processing, the z^{-N} in (33) can be eliminated because of the reset operation for every N cycles. The corresponding IIR structure to compute the k^{th} frequency component of the DCT is shown in Fig.10, in which

$$\Gamma_c(m) \triangleq (-1)^k C(k) \cos m\omega_k. \quad (34)$$

Once the last serial input $x(t)$ is fed into the module, the DCT coefficients can be obtained at the module outputs in parallel. The resulting parallel architecture is regular, modular, and fully-pipelined. Also, the SIPO feature can avoid the input buffers as well as the index mapping operation that are required in most PIPO DCT architectures [11][14]. One disadvantage of the IIR structure in Fig.10 is that the operation speed is constrained by the recursive loops. In what follows, we will reformulate the transfer function using the multirate approach so that speed constraint can be alleviated.

3.2 Low-Power Design of the IIR DCT

Splitting the input data sequence into the *even* sequence

$$x_e(t, n) = x(t + 2n - N + 1), \quad n = 0, 1, \dots, N/2 - 1, \quad (35)$$

and the *odd* sequence

$$x_o(t, n) = x(t + 2n - N + 2), \quad n = 0, 1, \dots, N/2 - 1, \quad (36)$$

(32) becomes

$$X_{DCT,k}(t) = C(k) \sum_{n=0}^{N/2-1} \cos\left[(4n+1)\frac{k\pi}{2N}\right] x_e(t, n) + C(k) \sum_{n=0}^{N/2-1} \cos\left[(4n+3)\frac{k\pi}{2N}\right] x_o(t, n). \quad (37)$$

Taking the z -transform on both sides of (37) and rearranging, we have

$$X_{DCT,k}(z) = \frac{C(k)((-1)^k - z^{-N/2})}{1 - 2 \cos 4\omega_k z^{-1} + z^{-2}} \left([X_e(z) - X_o(z)z^{-1}] \cos 3\omega_k + [X_o(z) - X_e(z)z^{-1}] \cos \omega_k \right) \quad (38)$$

where $X_e(z)$ and $X_o(z)$ are the z -transforms of $x_e(t, n)$ and $x_o(t, n)$, respectively. The parallel architecture to realize (38) is depicted in Fig.11. The *common circuit* at the left-hand side decimates the input serial data into the even and odd sequences and generates the common inputs for the module array. The numerator part and the denominator part of (38) are realized by the FIR structure and the IIR structure inside each DCT module at different index k . The overall architecture requires $(3N - 3)$ multipliers and $(3N + 1)$ adders plus a decimation circuit. Compared with the IIR DCT structure in Fig.10, this multirate DCT structure needs only $(N - 1)$ extra multipliers and $(N + 1)$ extra adders.

To achieve downsampling by the factor of four, we can split the input data sequence into four decimated sequences

$$g_i(t, n) \triangleq x(t + (4n + i) - N + 1), \quad i = 0, 1, 2, 3, \quad (39)$$

for $n = 0, 1, \dots, N/4 - 1$. Following the derivations in (37)-(38), we can write $X_{DCT,k}(z)$ as

$$\begin{aligned} X_{DCT,k}(z) &= \frac{C(k)((-1)^k - z^{-N/4})}{1 - 2 \cos 8\omega_k z^{-1} + z^{-2}} \left([G_0(z) - G_3(z)z^{-1}] \cos 7\omega_k + [G_1(z) - G_2(z)z^{-1}] \cos 5\omega_k \right. \\ &\quad \left. + [G_2(z) - G_1(z)z^{-1}] \cos 3\omega_k + [G_3(z) - G_0(z)z^{-1}] \cos \omega_k \right). \end{aligned} \quad (40)$$

where $G_i(z)$ is the z -transform of $g_i(t, n)$, $i = 0, 1, 2, 3$. The corresponding multirate architecture is shown in Fig.12.

From Fig.11 and Fig.12, we can see that basically the multirate DCT architectures retain all ad-

vantages of the original IIR structure in [3] such as modularity, regularity, and local interconnections. It is also interesting to note that the increase in hardware overhead grows only locally rather than globally, and the DCT architecture with $M = 4$ can be generated by reusing the modules in the $M = 2$ design (e.g., the FIR/IIR structures and the lattice structure in the common circuit). Therefore, neither global routing nor new module design is required in the $M = 4$ case. The characteristics of scalability, modularity, and local interconnections make the multirate structures very suitable for VLSI implementation. Unlike most PIPO DCT algorithms in which the interconnections will take up much of the chip area, the feature of local communications of our design can greatly reduce the power dissipation in the routing area. From the discussions in Section 2.3, it can be shown that the total power consumption for the multirate 16-point DCT can be reduced to $0.29P_0$ for $M = 2$, and $0.11P_0$ for $M = 4$, respectively. The significant power savings for the design with $M = 4$ is achieved only at the cost of $3N - 3$ extra multipliers and $3N + 3$ extra adders.

3.3 Low-Power Design of the IIR IDCT

The IIR transfer function for the block IDCT is given by [3]

$$H_{IDCT,n}(z) = \frac{(-1)^n C(1) \sin \omega_n}{1 - 2 \cos \omega_n z^{-1} + z^{-2}} + (C(0) - C(1))z^{-(N-1)}, \quad (41)$$

where $\omega_n \triangleq \frac{2n+1}{2N}\pi$. As with the derivations of the low-power IIR DCT, the multirate transfer function for the IDCT with $M = 2$ can be derived as

$$\begin{aligned} X_{IDCT,n}(z) &= \frac{(-1)^n C(1)}{1 - 2 \cos 2\omega_n z^{-1} + z^{-2}} \left(X_e(z) \sin 2\omega_n + (1 + z^{-1})X_o(z) \sin \omega_n \right) \\ &+ (C(0) - C(1))z^{-(N-1)}X(z). \end{aligned} \quad (42)$$

Similarly, the transfer function for $M = 4$ is

$$\begin{aligned} X_{IDCT,n}(z) &= \frac{(-1)^n C(1)}{1 - 2 \cos 4\omega_n z^{-1} + z^{-2}} \left(G_0(z) \sin 4\omega_n + [G_1(z) + G_3(z)z^{-1}] \sin 3\omega_n \right. \\ &+ (1 + z^{-1})G_2(z) \sin 2\omega_n + [G_3(z) + G_1(z)z^{-1}] \sin \omega_n \left. \right) \\ &+ (C(0) - C(1))z^{-(N-1)}X(z). \end{aligned} \quad (43)$$

The corresponding low-power IIR IDCT structures based on (42) and (43) are shown in Fig.13 and Fig.14, respectively, where the multiplier coefficient is defined as

$$\Gamma_s(m) \triangleq (-1)^n C(1) \sin m\omega_n. \quad (44)$$

As we can see, the low-power IDCT design has similar structures as the low-power DCT except little difference in the common circuit. Therefore, it is possible to integrate both the forward and backward transforms into one architecture by suitably multiplexing the data path in the common circuit and the coefficients inside the modules.

3.4 Polyphase Representation

In the preceding discussions, we have shown how to perform the multirate DCT/IDCT by rearranging the z -transforms of the decimated sequences. Here we will show a systematic way to derive the results by applying the polyphase decomposition to the original IIR transfer function.

Substitute the identity that

$$\frac{1}{1 - 2 \cos 2\omega_k z^{-1} + z^{-2}} = \frac{1 + 2 \cos 2\omega_k z^{-1} + z^{-2}}{1 - 2 \cos 4\omega_k z^{-2} + z^{-4}} \quad (45)$$

into the IIR DCT transfer function in (33). After rearranging, $H_{DCT,k}(z)$ under block processing can be written as

$$H_{DCT,k}(z) = \frac{(-1)^k C(k)}{D(z^2)} [H_0(z^2) + z^{-1} H_1(z^2)] \quad (46)$$

where

$$\begin{aligned} D(z^2) &= 1 - 2 \cos 4\omega_k z^{-2} + z^{-4}, \\ H_0(z^2) &= (\cos \omega_k - \cos 3\omega_k z^{-2}), \\ H_1(z^2) &= (\cos 3\omega_k - \cos \omega_k z^{-2}). \end{aligned} \quad (47)$$

(46) is the polyphase representation of $H_{DCT,k}(z)$ with $M = 2$, and its corresponding polyphase implementation is shown in Fig.15(a). The downsampling operation $\downarrow N$ at the right end denotes that we pick up the DCT coefficients at the N^{th} clock cycle and ignore all the previous intermediate results. Given this polyphase implementation, we can use the noble identities [5] to distribute the

downsampling operation towards the left and obtain Fig.15(b), which will lead to the multirate DCT architecture in Fig.11. Thus, we can process the input data at two times slower clock rate. After $N/2$ iterations, the DCT coefficients are available at the output ends. Similarly, $M = 4$ can be achieved by performing another polyphase decomposition on $\frac{1}{D(z^2)}$ in (46). After some algebraic simplifications, we can obtain (40) and its corresponding implementation allows us to operate at four times slower clock rate. The polyphase decomposition can also be used to derive the results for the multirate IDCT. In the companion paper [8], we will apply this methodology to obtain the low-power architecture of logarithmic complexity as well as the unified transformation module design.

4 Comparisons of Architectures

In this section, we would like to discuss the hardware complexity of the two algorithm-based low-power approaches (the Chebyshev polynomial approach and the polyphase decomposition approach) proposed in this paper. Also, we will compare the proposed multirate SIPO architectures with the existing SIPO and PIPO architectures [3][14]. Table 1 summarizes the hardware cost for all the proposed architectures under normal operation and under multirate operation ($M = 2, 4$). As we can see, the hardware overhead of the low-power design is linear complexity increase for the speed compensation. As to the two approaches (Chebyshev and polyphase), the Chebyshev IDCT requires $(N - 1)$ less multipliers than the IIR IDCT in both normal and multirate operations. This saving is in particular preferable for the applications which require cost-effective IDCT such as HDTV receivers. On the contrary, the Chebyshev DCT has almost the same complexity as the IIR DCT. Since the Chebyshev DCT needs one more iteration to finish the transform, the polyphase IIR DCT is a better choice for the implementations.

Next, we would like to compare our low-power DCT architecture with those proposed in [14] and [3]. The architecture in [14], which utilizes factorization method to perform fast DCT, is a typical representative of the PIPO fast algorithms. The IIR structure proposed in [3], on the other hand, is a good example of the SIPO algorithms. A comparison regarding their inherent properties is listed in Table 2. The advantages of the SIPO approach over the PIPO approach in their VLSI implementation, such as local communication and linear hardware complexity increase, have been discussed thoroughly in [2] and [3]. Nevertheless, when the speed compensation capability is of concern, the PIPO is also a good choice since the block PIPO processing with block size N is

equivalent to decimating the input data by a factor of N . However, this advantage is obtained at the price of globally increased hardware and routing paths. Besides, the block size is usually restricted to be power of two due to the “divide-and-conquer” nature of those PIPO fast algorithms. From Table 2, we can see that our multirate SIPO approach is a good compromise between the other two approaches. Basically, the multirate approach inherits all the advantages of the existing SIPO approach; meanwhile, it can compensate the speed penalty at the expense of “locally” increased hardware and routing, which is not the case in the PIPO approach. Although some restriction is imposed on the data size N due to the downsampling operation, *i.e.*,

$$N = Mk, \quad k \in Z^+ \quad (48)$$

(M is the decimation factor and Z^+ denotes any positive integer), the choice of N is much more flexible compared with the PIPO algorithms.

The other advantage of the SIPO approaches is in the computation of the pruning DCT [15]. In the DCT-based signal compression algorithms, the most useful information of the signal is kept in the low frequency DCT components. Therefore, retaining only $N_0 < N$ coefficients is sufficient for the lossy data compression. Although the pruning DCT can be computed from the PIPO DCT architecture by removing the unnecessary data paths and computational operators [15], the global communication is still the major drawback for its implementation as N increases. On the contrary, the SIPO architecture in [3] and our low-power design can be readily applied to the pruning DCT by simply implementing the first N_0 DCT modules for the computation of the first N_0 DCT coefficients.

5 Conclusions

In this paper, we presented the algorithm-based low-power design of the transform coding kernels based on the multirate approach. We have shown that by either exploiting the properties of the Chebyshev polynomial or reformulating the IIR DCT/IDCT algorithms, we can reduce the operating frequency of the transforms at the architectural level without degrading the system throughput rate. Such compensation capability will lead to drastic savings in the total power consumption. Therefore, the proposed low-power transform coding kernels will be effective for the low-power/high-performance signal processing systems.

It is worth mentioning that since the multirate approach is derived at the word level, other arithmetical-level techniques, such as bit-level downsampling [16] and distributed arithmetic [17][18], can be employed in the VLSI implementation to further reduce the power consumption. In general, they do not explicitly exploit the inherent properties of the orthogonal transforms. As a result, they achieve the speed compensation at the arithmetical level rather than at the algorithmic level.

The other attractive application of our design is in the very high-speed signal processing. Presently, in most VLSI implementation of the orthogonal transforms, the input data rate is limited by the speed of the adders and multipliers in the circuit. In the video-rate applications such as HDTV, the speed constraint will result in the use of expensive high-speed multiplier/adder circuits or full-custom design. Thus, the manufacturing cost as well as the design cycle will increase drastically. By employing the multirate parallel architectures discussed in this paper, the speed constraint can be resolved at the architectural level with the same design environment and fabrication technology. For example, if we want to perform DCT for serial data at 200 MHz, we may use the parallel architecture in Fig.12, in which only 50MHz adders and multipliers are required. Therefore, we can perform very high-speed DCT by using only low-cost and low-speed processing elements.

References

- [1] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp. 473–484, April 1992.
- [2] K. J. R. Liu and C. T. Chiu, "Unified parallel lattice structures for time-recursive Discrete Cosine/Sine/Hartley transforms," *IEEE Trans. Signal Processing*, vol. 41, pp. 1357–1377, March 1993.
- [3] K. J. R. Liu, C. T. Chiu, R. K. Kolagotla, and J. F. J. Ja', "Optimal unified architectures for the real-time computation of time-recursive discrete sinusoidal transforms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 168–180, April 1994.
- [4] G. Steidl, "Chebyshev polynomial derivation of composite-length DCT algorithms," *Signal Processing*, vol. 29, pp. 17–27, Oct. 1992.
- [5] P. P. Vaidyanathan, *Multirate systems and filter banks*. Englewood Cliffs, NJ: Prentice Hall, 1993.
- [6] D. A. Pucknell and K. Eshraghian, *Basic VLSI Design- Systems and Circuits*. Prentice Hall, 2nd ed., 1988.
- [7] K. Konstantinides, "Fast subband filtering in MPEG audio coding," *IEEE Signal Processing Letters*, vol. 1, pp. 26–28, Feb. 1994.

- [8] A.-Y. Wu and K. J. R. Liu, "Algorithm-based low-power transform coding architectures- Part II: Logarithmic complexity, unified architecture, and finite-precision analysis," *submit to IEEE Trans. Circuits Syst. II: Analog and Digital Signal Processing*, 1995.
- [9] L. Fox and I. B. Parker, *Chebyshev polynomials in numerical analysis*. London: Oxford University Press, 1968.
- [10] H. R. Schwarz, *Numerical analysis : A comprehensive introduction*. John Wiley & Sons, 1989.
- [11] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 35, pp. 1455–1461, Oct. 1987.
- [12] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design- A Systems Perspective*. Addison-Wesley, 1985.
- [13] J.-G. Chung and K. K. Parhi, "Pipelining of lattice IIR digital filters," *IEEE Trans. Signal Processing*, vol. 42, pp. 751–761, April 1994.
- [14] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. 32, pp. 1243–1245, Dec. 1984.
- [15] A. N. Skodras, "Fast discrete cosine transform pruning," *IEEE Trans. Signal Processing*, vol. 42, pp. 1833–1837, July 1994.
- [16] S. Cucchi and M. Fratti, "A novel architecture for VLSI implementation of the 2-D DCT/IDCT," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, pp. V.693–696, 1992.
- [17] M. T. Sun, T. C. Chen, and A. M. Gottlieb, "VLSI implementation of a 16×16 discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 610–617, Apr. 1989.
- [18] S. Wolter, D. Birreck, and R. Laur, "Classification for 2D-DCTs and a new architecture with distributed arithmetic," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2204–2207, 1991.

	Normal Operation		Downsampling by 2 ($M = 2$)		Downsampling by 4 ($M = 4$)		Extra iteration
	Multiplier	Adder	Multiplier	Adder	Multiplier	Adder	
Chebyshev DCT	$2N - 2$	$3N - 1$	$3N - 3$	$4N$	$5N - 5$	$6N + 3$	Yes
Chebyshev IDCT	$N + 2$	$3N$	$2N + 2$	$4N + 1$	$4N + 2$	$6N + 4$	No
IIR DCT	$2N - 2$	$2N$	$3N - 3$	$3N + 1$	$5N - 5$	$5N + 3$	No
IIR IDCT	$2N + 1$	$3N$	$3N + 1$	$4N + 1$	$5N + 1$	$6N + 2$	No

Table 1: Comparison of hardware cost for the DCT and IDCT architectures with their low-power designs in terms of 2-input multipliers and 2-input adders.

	Liu <i>et. al.</i> [3]	Proposed multirate IIR DCT with $M = 4$	Lee [14]
Data processing rate	f_s	f_s/M	f_s/N
No. of Multipliers	$2N - 2$	$(M + 1)N$ (in order)	$\left(\frac{3N}{2}\right) \log_2 N$ (in order)
No. of Adders	$2N$	$(M + 1)N$ (in order)	$\left(\frac{N}{2}\right) \log_2 N$
Latency	N	N	$[\log_2 N(\log_2 N - 1)]/2$
Restriction on transform size N	No	$Mk, k \in Z^+$	$2^k, k \in Z^+$
Requirement for input buffer	No	No	Yes
Index mapping	No	No	Yes
Communication	Local	Local	Global
I/O operation	SIPO	SIPO	PIPO
Speed compensation capability	N/A	Good (at the expense of locally increased hardware overhead and local routing)	Good (at the expense of globally increased hardware overhead and global routing)
Power consumption in routing	Negligible	Negligible	Noticeable as N increases
Application to pruning DCT	Direct	Direct	Needs many modifications and global interconnections

Table 2: Comparisons of different DCT architectures, where f_s denotes the data sample rate, M denotes the programmable downsampling factor, and N is the block size.

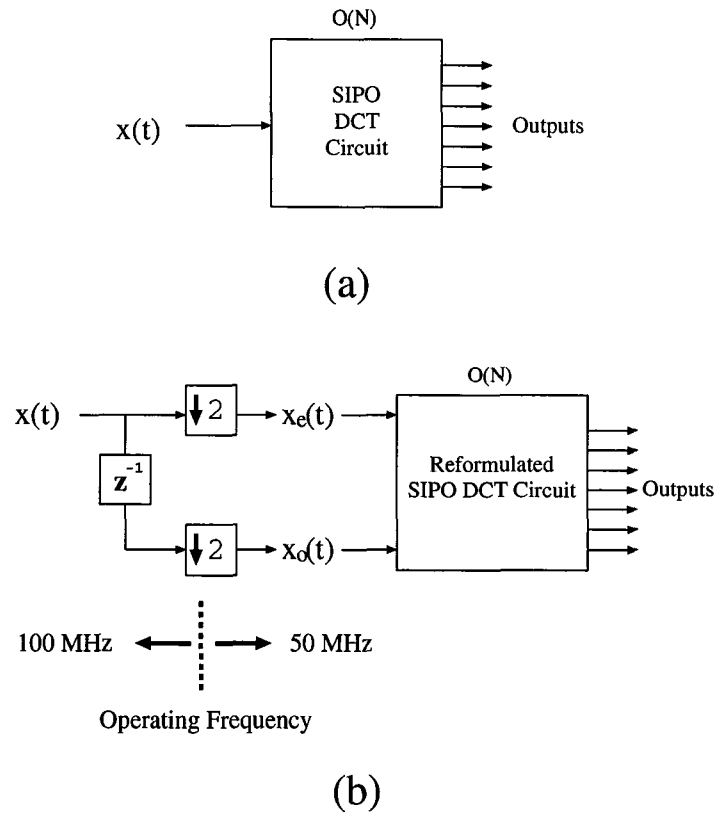


Figure 1: (a) Original SIPO DCT circuit. (b) Low-power DCT circuit using the multirate approach.

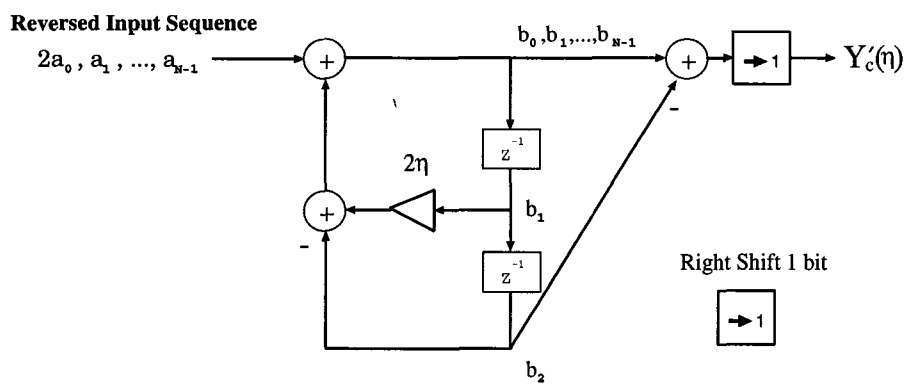


Figure 2: Recursive architecture to evaluate $Y'_c(\eta)$.

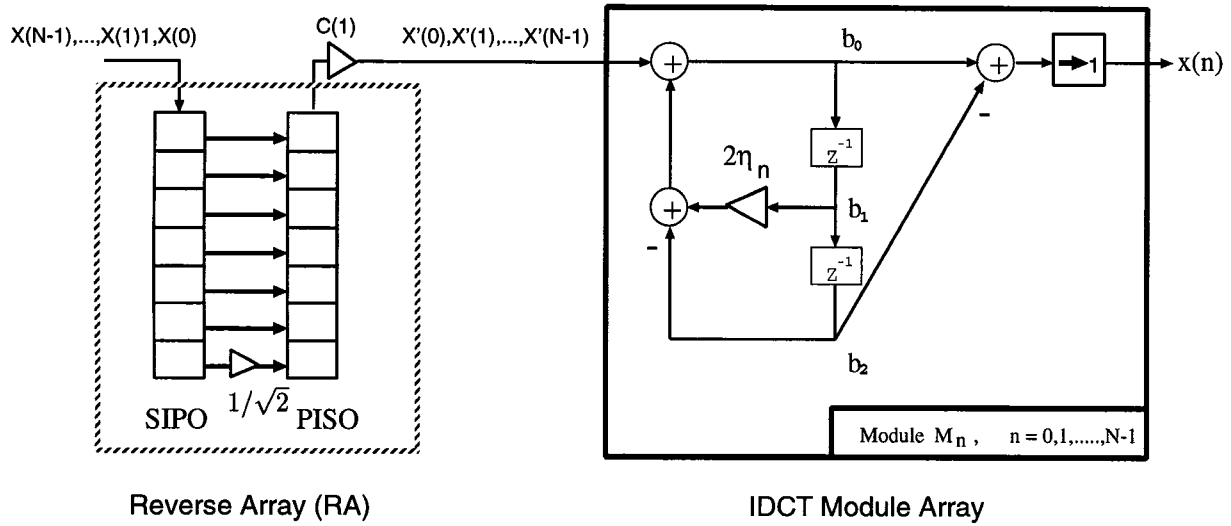


Figure 3: Parallel Chebyshev IDCT architecture.

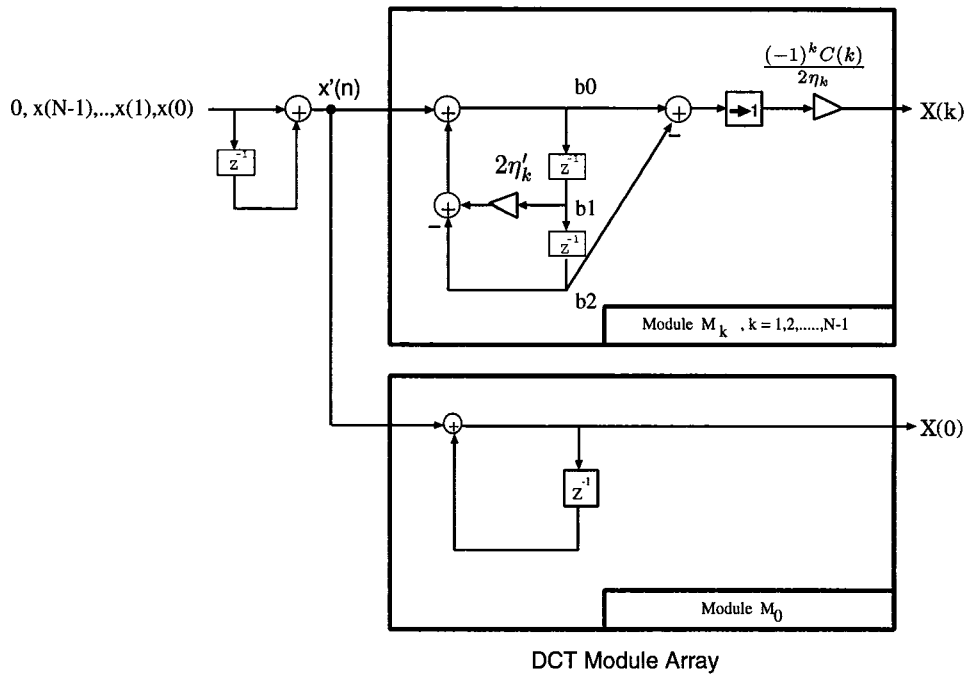


Figure 4: Parallel Chebyshev DCT architecture.

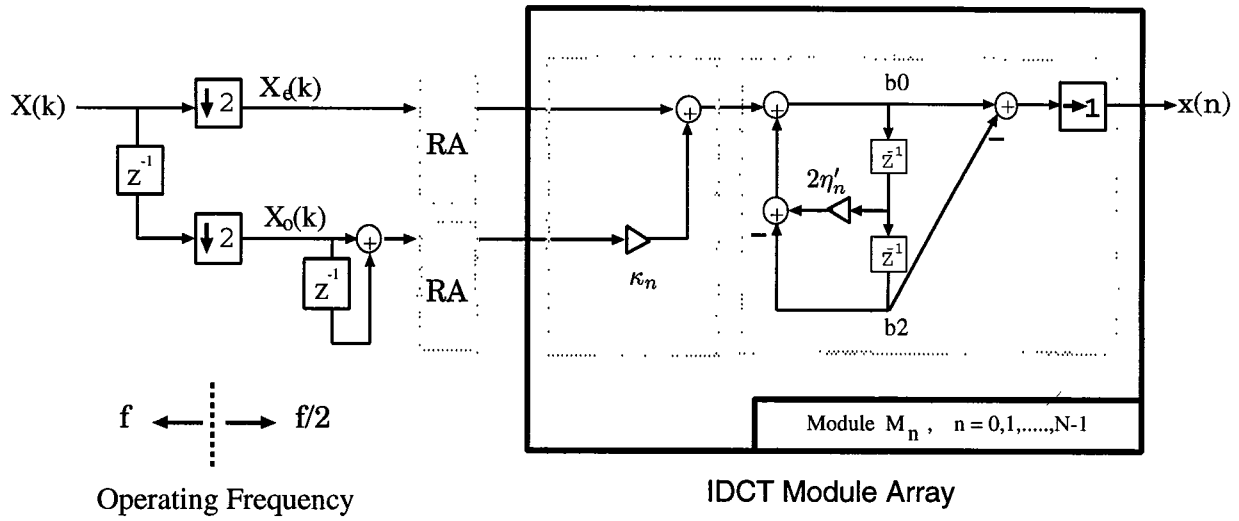


Figure 5: Low-power parallel Chebyshev IDCT architecture with decimation factor of two.

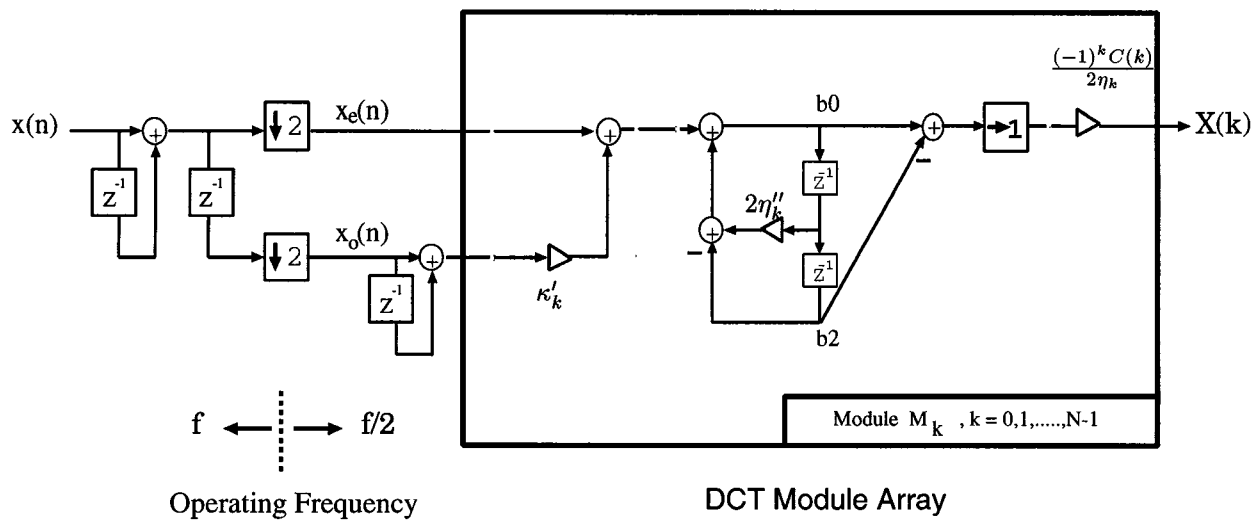


Figure 6: Low-power parallel Chebyshev DCT architecture with decimation factor of two.

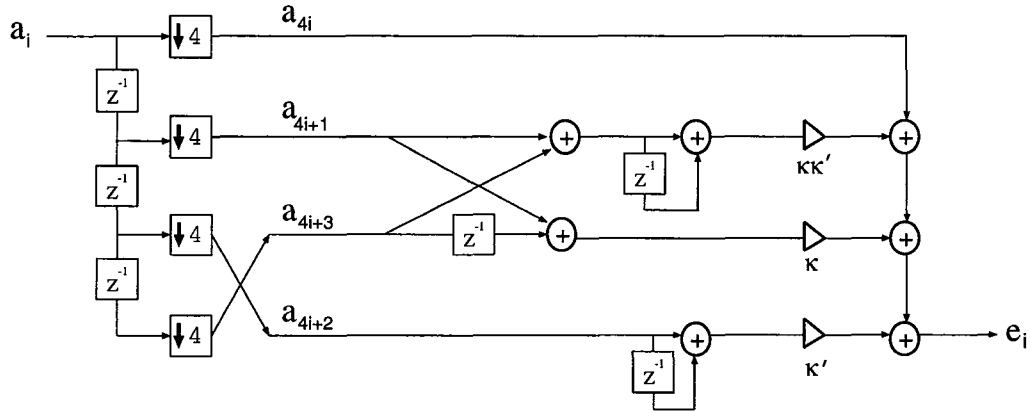


Figure 7: Evaluation of e_i using the downsampling circuit.

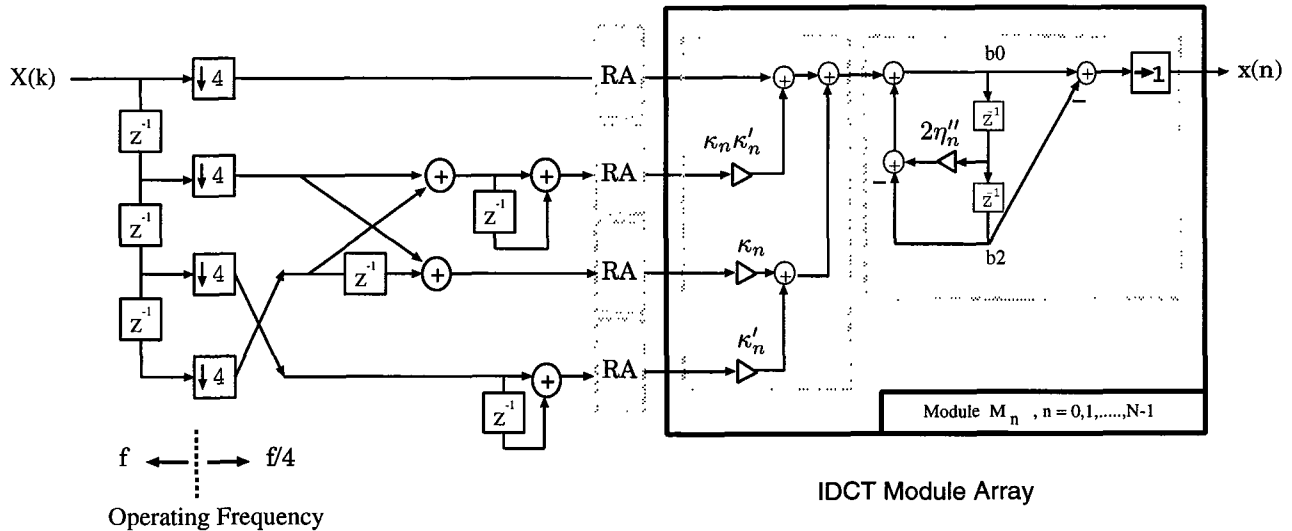


Figure 8: Low-power parallel Chebyshev IDCT architecture with decimation factor of four, where $\eta''_n = 2(\eta'_n)^2 - 1$ and $\kappa'_n = 1/(2\eta'_n)$.

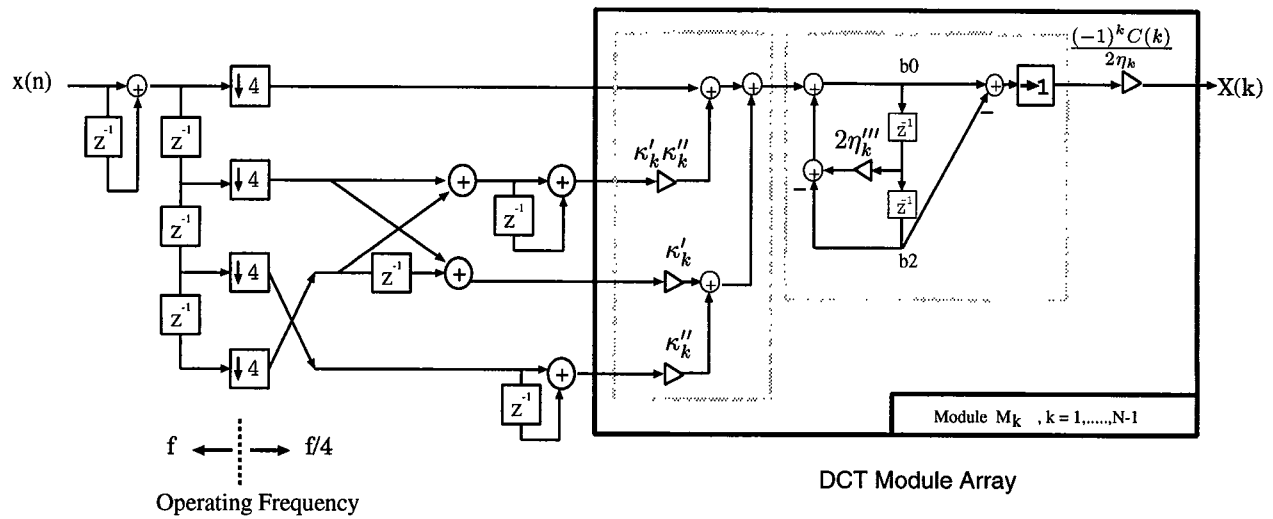


Figure 9: Low-power parallel Chebyshev DCT architecture with decimation factor of four, where $\kappa''_k = 1/\eta''_k$ and $\eta'''_k = 2(\eta''_k)^2 - 1$.

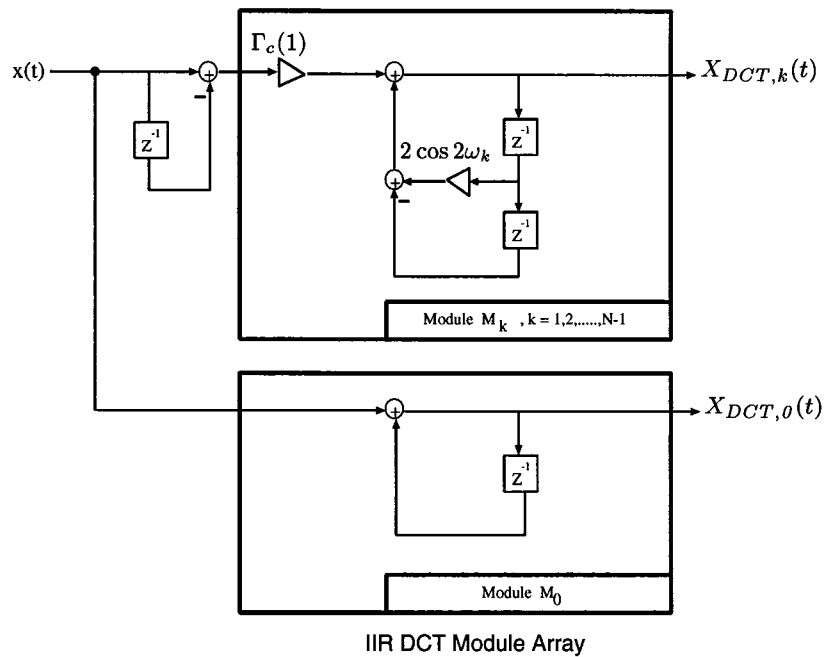


Figure 10: IIR DCT architecture.

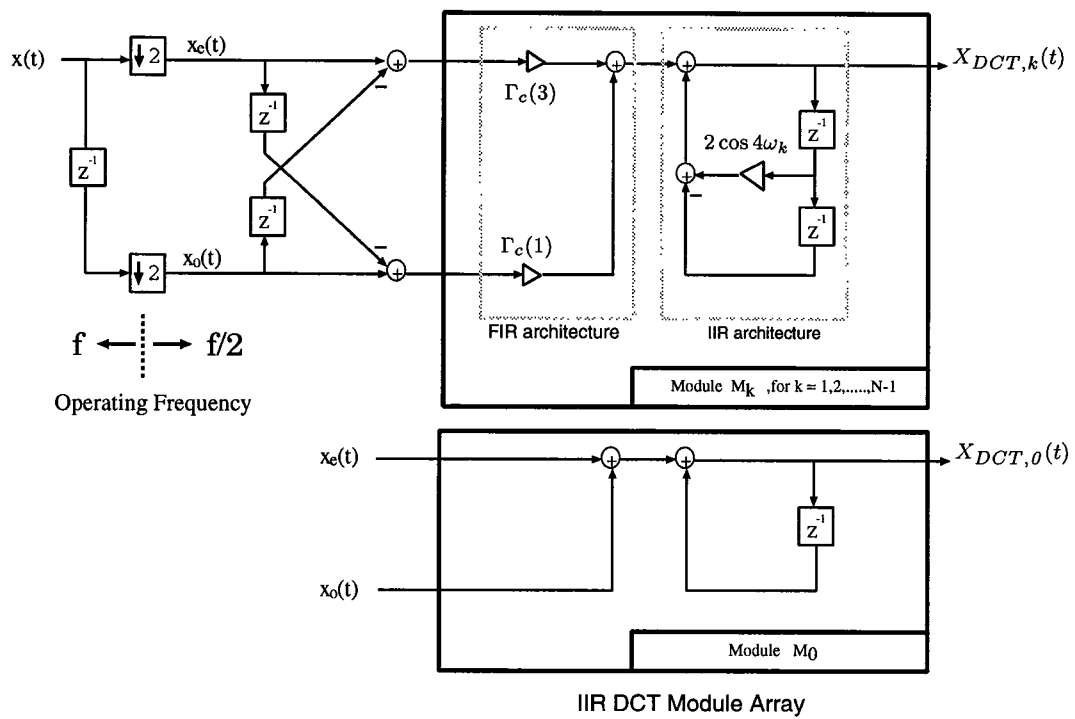


Figure 11: Low-power polyphase IIR DCT architecture with $M = 2$.

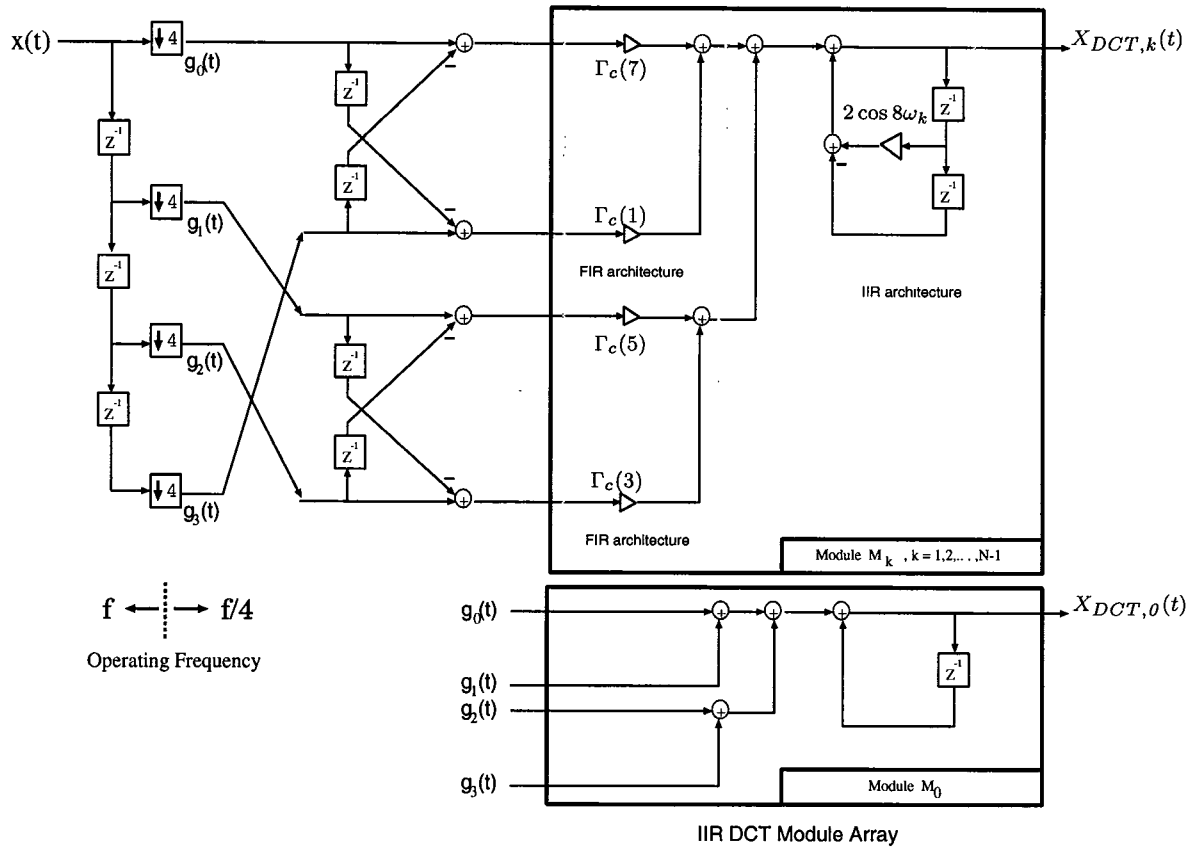


Figure 12: Low-power polyphase IIR DCT architecture with $M = 4$.

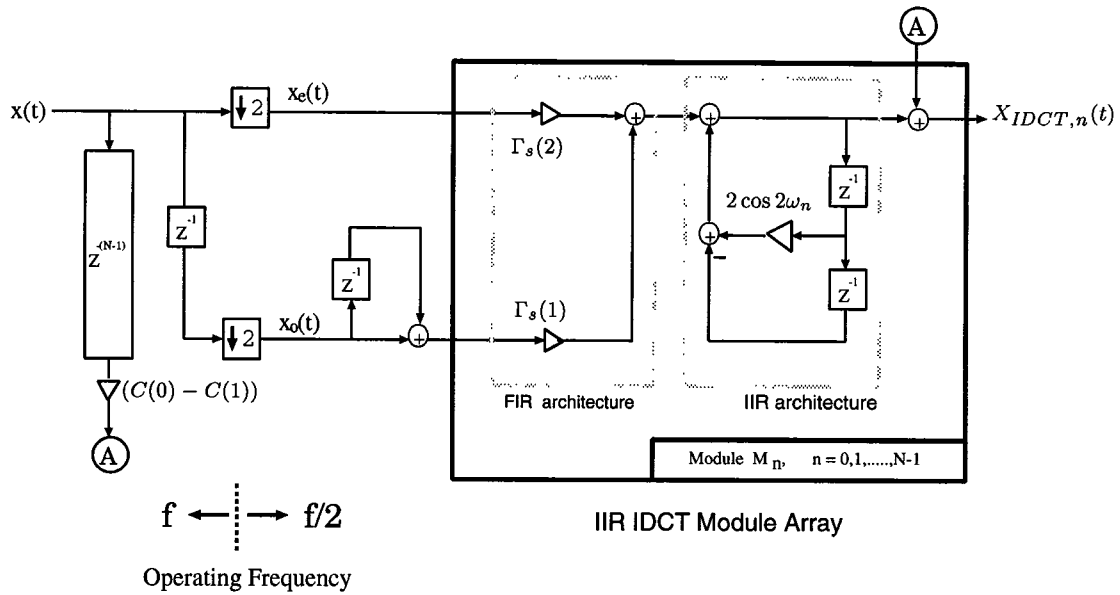


Figure 13: Low-power polyphase IIR IDCT architecture with $M = 2$.

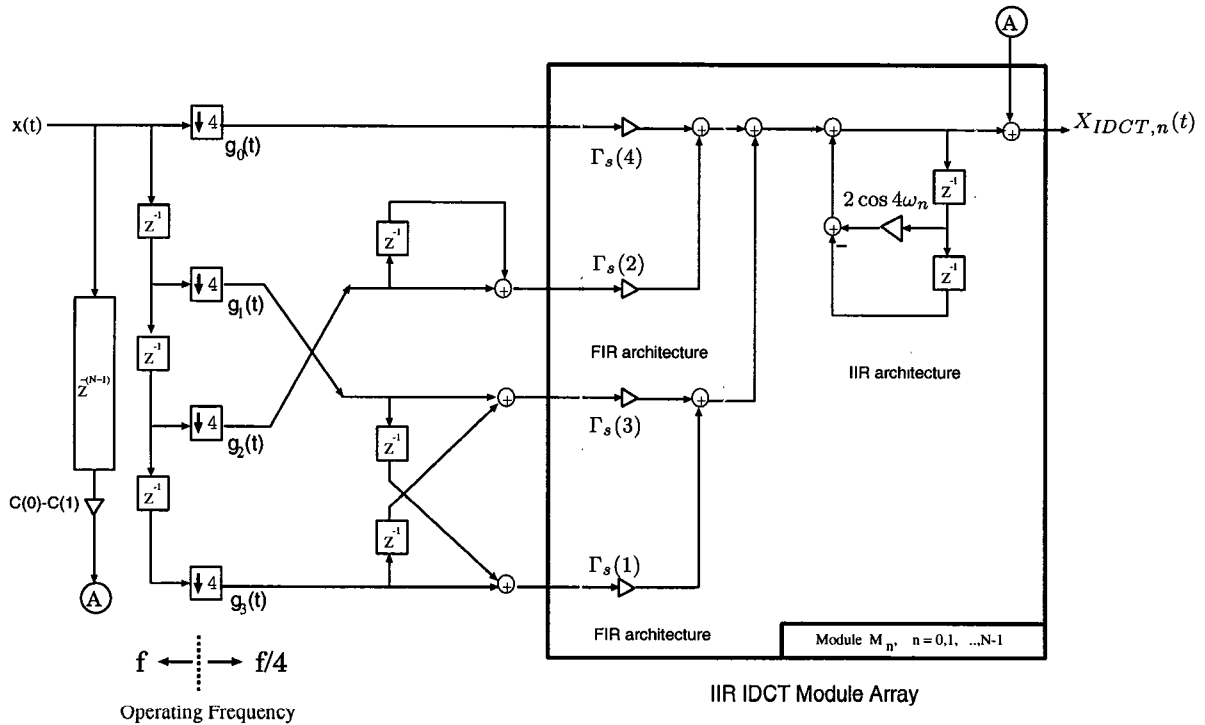


Figure 14: Low-power polyphase IIR IDCT architecture with $M = 4$.

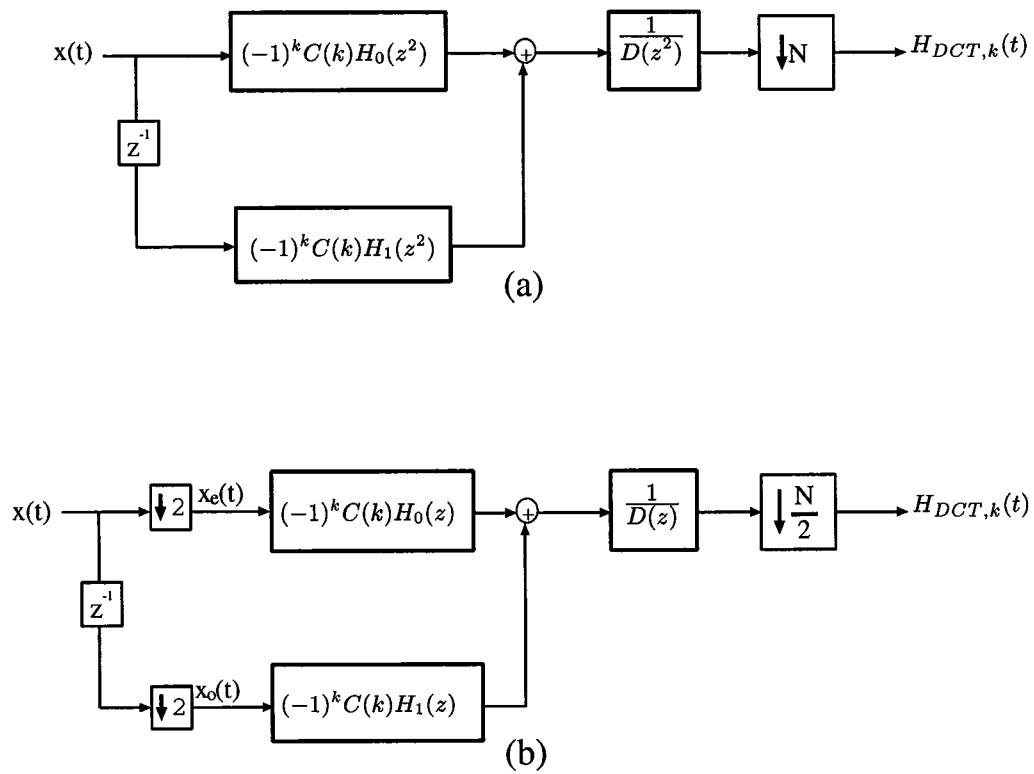


Figure 15: (a) Polyphase representation of $H_{DCT,k}(z)$. (b) Polyphase representation of $H_{DCT,k}(z)$ after applying the noble identity.