

# Efficient Recursive Algorithm for the Inverse Discrete Cosine Transform

Lap-Pui Chau and Wan-Chi Siu

**Abstract**—Recursive algorithms have been found very effective for realization using software and very large scale integrated circuit (VLSI) techniques. Recently, some recursive algorithms have been proposed for the realization of the inverse discrete cosine transform (IDCT). In this paper, an efficient recursive algorithm for the IDCT with arbitrary length is presented. By using some appropriate iterative techniques, the formulation of the IDCT can be implemented effectively using recursive equations, and the hardware complexity is further reduced as compared with the approaches in the literature.

**Index Terms**—Discrete cosine transform (DCT), recursive algorithm.

## I. INTRODUCTION

THE discrete cosine transform (DCT) [1] is widely used in the area of digital signal processing. Many algorithms for the computation of the DCT have been proposed. It is significant to point out that in many applications, the efficiency of the realization of the inverse DCT (IDCT) is even more important compared to that of the DCT. Hence, we concentrate on the realization of the IDCT in this paper. Chau and Siu [2] proposed a recursive algorithm for computing the DCT, the structure is regular and parallel, which is suitable for both software and VLSI implementations. Recently, Wang, Jullien, and Miller [3], and Aburdene, Zheng, and Kozick [4] suggested improved versions of this recursive algorithm for the DCT and the IDCT. The DCT requires less multiplier as compared with that of the Chau and Siu's algorithm. In this paper, a more efficient recursive IDCT algorithm is proposed, which reduces one multiplier as compared with the existing IDCT algorithms in [3] and [4]. A significant reduction in the number of gate count for VLSI implementation can hence be achieved.

## II. DERIVATION OF ALGORITHM

Let us recall a recursive kernel that was suggested by Chau and Siu [2] for the realization of the DCT:

$$\begin{aligned} V_m \sin x \\ = \sum_{k=m}^{N-1} Y(k) * \sin(k-m+1)x \end{aligned}$$

Manuscript received April 5, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Y. Hua.

L.-P. Chau is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: lpchau@ieee.org).

W.-C. Siu is with the Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong.

Publisher Item Identifier S 1070-9908(00)08359-0.

$$\begin{aligned} &= Y(m) * \sin x + \sum_{k=m+1}^{N-1} Y(k) * \sin(k-m+1)x \\ &= Y(m) * \sin x + \sum_{k=m+1}^{N-1} Y(k) * (2 \cos x * \sin(k-m)x \\ &\quad - \sin(k-m-1)x) \\ &= Y(m) * \sin x + 2 \cos x \sum_{k=m+1}^{N-1} Y(k) * \sin(k-m)x \\ &\quad - \sum_{i=m+2}^{N-1} Y(k) * \sin(k-m-1)x \\ &= Y(m) * \sin x + 2 \cos x * V_{m+1} \sin x - V_{m+2} \sin x. \quad (1) \end{aligned}$$

Hence, we have

$$\begin{aligned} V_m &= Y(m) + 2 \cos x * V_{m+1} - V_{m+2} \\ &\quad \text{for } m = N-1, \dots, 1, 0. \quad (2) \end{aligned}$$

The IDCT of a sequence  $\{Y(k): k = 0, 1, \dots, N-1\}$  can be written as

$$\begin{aligned} y(i) &= \sum_{k=0}^{N-1} Y(k) * \cos\left(\frac{\pi}{2N} * (2i+1) * k\right) \\ &\quad \text{for } i = 0, 1, \dots, N-1. \quad (3) \end{aligned}$$

We intend to obtain a simplified expression for the IDCT. Let  $x = \pi * (2i+1)/2N$ , then

$$\begin{aligned} y(i) &= \sum_{k=0}^{N-1} Y(k) * \cos(x * k) \\ &= Y(0) + \sum_{k=1}^{N-1} Y(k) * \cos(x * k). \quad (4) \end{aligned}$$

Our objective is to apply the recursive kernel into (4). Hence, we multiply both sides by  $\sin x$

$$\begin{aligned} &y(i) * \sin x \\ &= Y(0) * \sin x + \sum_{k=1}^{N-1} Y(k) * \cos(x * k) * \sin x \\ &= Y(0) * \sin x + \sum_{k=1}^{N-1} Y(k) (\cos(x * k) * \sin x \\ &\quad - \sin(x * k) * \cos x + \sin(x * k) * \cos x) \\ &= Y(0) * \sin x - \sum_{k=1}^{N-1} Y(k) * \sin(k-1)x \end{aligned}$$

$$\begin{aligned}
 & + \sum_{k=1}^{N-1} Y(k) * \sin(x * k) * \cos x \\
 & = Y(0) * \sin x + \sum_{k=1}^{N-1} Y(k) * \sin(x * k) * \cos x \\
 & \quad - \sum_{k=2}^{N-1} Y(k) * \sin(k-1)x.
 \end{aligned} \tag{5}$$

By substituting (1) into (5), we have

$$\begin{aligned}
 y(i) * \sin x \\
 & = Y(0) * \sin x + \sin x * \cos x * V_1 - \sin x * V_2.
 \end{aligned} \tag{6}$$

Divide both sides by  $\sin x$ , and we finally obtain

$$\begin{aligned}
 y(i) & = Y(0) + \cos x * V_1 - V_2 \\
 & = V_0 - V_1 \cos x \\
 & = V_0 - V_1 \cos\left(\frac{\pi(2i+1)}{2N}\right).
 \end{aligned} \tag{7}$$

Equation (7) is the simplest formulation to obtain the final result, while the recursive kernel is obtained by substituting  $x = \pi * (2i+1)/2N$  into (2), which can be used for the realization of the IDCT with an extremely simple hardware structure.

From (7), it appears that one multiplication is required for computing  $y(i)$ . Fortunately, the multiplication  $V_1 * \cos(\pi(2i+1)/2N)$  can be eliminated by using the existing multiplier in the recursive kernel, which can be obtained by “divided by two” from the output of multiplier  $2 \cos(\pi(2i+1)/2N) * V_{m+1}$ . The “divided by two” operation is actually a right shift one bit process that can be easily implemented by redefining the bus connection. Fig. 1 shows the block diagram for the realization of the IDCT using the recursive filter structure, which requires one less multiplier as

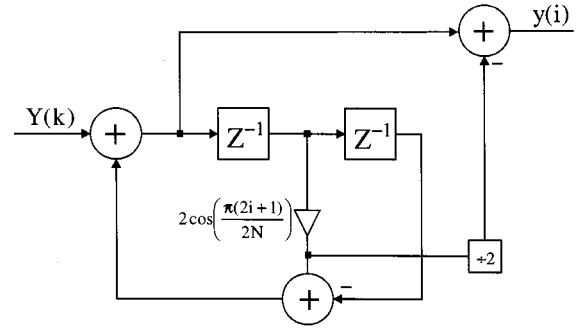


Fig. 1. Recursive filter structure for computing IDCT.

compared to that of the IDCT structures proposed by Wang *et al.* [3], and Aburdene *et al.* [4]. The delay elements of the recursive filter must be reset to zero before recursion.

### III. CONCLUSION

An efficient recursive algorithm for the implementation of the IDCT has been derived. The algorithm is suitable for both software and hardware realization. A significant reduction in terms of multiplier is achievable for VLSI realization of the IDCT using the present formulation.

### REFERENCES

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Trans. Comput.*, vol. 23, pp. 90–93, Jan. 1974.
- [2] L. P. Chau and W. C. Siu, “Recursive algorithm for the discrete cosine transform with general lengths,” *Electron. Lett.*, vol. 30, no. 3, pp. 197–198, Feb. 1994. *Errata: Electron. Lett.*, vol. 30, no. 7, pp. 608, 1994.
- [3] Z. Wang, G. A. Jullien, and W. C. Miller, “Recursive algorithms for the forward and inverse discrete cosine transform with arbitrary length,” *IEEE Signal Processing Lett.*, vol. 1, pp. 101–102, July 1994.
- [4] M. F. Aburdene, J. Zheng, and R. J. Kozick, “Computation of discrete cosine transform using Clenshaw’s recurrence formula,” *IEEE Signal Processing Lett.*, vol. 2, pp. 155–156, Aug. 1995.