

A REFINED FAST 2-D DISCRETE COSINE TRANSFORM ALGORITHM WITH REGULAR BUTTERFLY STRUCTURE

Yuh-Ming Huang^{1,2}, Ja-Ling Wu^{1,2}, and Chiou-Ting Hsu¹

¹Department of Computer Science and Information Engineering
National Taiwan University, Taipei, 107

²Department of Information Engineering, National Chi-Nan University, Puli, 545
Taiwan, Republic of China

Abstract

In this paper, a fast computation algorithm for the two-dimensional discrete cosine transform (2-D DCT) is derived based on index permutation. As a result, only the computation of N N -point 1-D DCT's and some post-additions are required for the computation of an $(N \times N)$ -point 2-D DCT. Furthermore, as compared with [7], the derivation of the refined algorithm is more succinct, and the associated post-addition stage possesses a more regular butterfly structure. The regular structure of the proposed algorithm makes it more suitable for VLSI and parallel implementations.

I. Introduction

Since DCT approaches the statistically optimal Karhunen-Loeve transform (KLT) for highly correlated signals. It has found wide applications in speech and image processing as well as telecommunication signal processing for the purpose of data compression, feature extraction, image reconstruction, and filtering. Thus, many algorithms and VLSI architectures for the fast computation of DCT have been proposed [1]-[6].

Among those algorithms, [5] and [6] are believed to be the most efficient 2-D DCT algorithms in the sense of minimizing any measure of computational complexity. However, the main drawbacks of these algorithms are the requirements of complex computation for [5] and complicated matrix decomposition for [6]. Moreover, the non-modularized structure of these algorithms may complicate the design and control of the concurrent VLSI implementation.

Recently, Cho and Lee [7] proposed a fast modularized DCT algorithm, in which an $(N \times N)$ -point 2D DCT could be obtained by computing N N -point 1-D DCT's and a post-addition stage. In a later work [9], they also provide regular expressions for the input-output relations of the post-addition stage. However, the number of required additions is increased as an expense for improving the regularity in the structure.

Based on the idea of [7], in this paper, an index permutation based algorithm for computing the 2-D DCT is proposed. Although the resultant computational complexity is the same as that of [7], the derivation of the refined algorithm is more succinct, and the post-addition stage of the refined algorithm has a more regular butterfly structure.

II. The Refined Fast Algorithm for Computing the 2-D DCT

For a given input data sequence $f_{i,j}$, $0 \leq i \leq N-1$, $0 \leq j \leq N-1$, the 2-D DCT and its inverse are given by [1]

$$F_{m,n} = \frac{2}{N} c_m c_n \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{i,j} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N}, \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (1-1)$$

and

$$f_{i,j} = \frac{2}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} c_m c_n F_{m,n} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N}, \quad 0 \leq i \leq N-1, 0 \leq j \leq N-1, \quad (1-2)$$

where

$$c_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k = 0 \\ 1, & \text{otherwise} \end{cases}. \quad (1-3)$$

For convenience, the normalization factors c_m and c_n are not included in the following derivations. Thus, the denormalized 2-D DCT can be expressed as

$$Y_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f_{i,j} \cos \frac{(2i+1)m\pi}{2N} \cos \frac{(2j+1)n\pi}{2N}, \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (2)$$

and $F_{m,n} = (2/N) c_m c_n Y_{m,n}$.

After some permutation of the input data sequence [2], eqn. (2) can be written as

$$Y_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+1)m\pi}{2N} \cos \frac{(4j+1)n\pi}{2N}, \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (3)$$

where

$$X_{i,j} = \begin{cases} f_{2i,2j} & 0 \leq i,j \leq N/2-1 \\ f_{2i,2N-2j-1} & 0 \leq i \leq N/2-1, N/2 \leq j \leq N-1 \\ f_{2N-2i-1,2j} & N/2 \leq i \leq N-1, 0 \leq j \leq N/2-1 \\ f_{2N-2i-1,2N-2j-1} & N/2 \leq i,j \leq N-1. \end{cases} \quad (4)$$

Based on the idea of [7], let

$$A_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+1)m + (4j+1)n}{2N} \pi,$$

$$0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (5-1)$$

and

$$B_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,j} \cos \frac{(4i+1)m - (4j+1)n}{2N} \pi, \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (5-2)$$

then

$$Y_{m,n} = (A_{m,n} + B_{m,n})/2. \quad (6)$$

Since $4i+1$ and N are coprime to each other, i.e. $(4i+1, N)^1 = 1$, the permutation $(4i+1)j+i$ modulo N maps all values of j . Let q_{ij} be the quotient of $(4i+1)j+i$ divided by N . Hence, the kernels of the 2-D transforms represented in eqn. (5-1) and eqn. (5-2) can be rewritten as 1-D DCT's by replacing j with $(4i+1)j+i-Nq_{ij}$. That is,

$$A_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,((4i+1)j+i)_{\%N}} \cos \frac{(4i+1)(m+(4j+1)n) - 4Nq_{ij}n}{2N} \pi$$

$$= \sum_{j=0}^{N-1} \left[\sum_{i=0}^{N-1} X_{i,((4i+1)j+i)_{\%N}} \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi \right] \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1, \quad (7-1)$$

and

$$B_{m,n} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{i,((4i+1)j+i)_{\%N}} \cos \frac{(4i+1)(m-(4j+1)n) + 4Nq_{ij}n}{2N} \pi \\ = \sum_{j=0}^{N-1} \left[\sum_{i=0}^{N-1} X_{i,((4i+1)j+i)_{\%N}} \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \right] \quad 0 \leq m \leq N-1, 0 \leq n \leq N-1. \quad (7-2)$$

where $\langle x \rangle_N$ denotes x modulo N .

For the simplicity of notation, $X_{i,((4i+1)j+i)_{\%N}}$ is denoted as $\bar{X}_{j,i}$. Then, it can be seen that the 2-D input data sequence is grouped into N distinct data sets of size N , that is

$$\left\{ \left(\bar{X}_{j,i}, 0 \leq i \leq N-1 \right), 0 \leq j \leq N-1 \right\}, \text{ and}$$

the equations

$$\sum_{i=0}^{N-1} \bar{X}_{j,i} \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi \quad (8-1)$$

and

$$\sum_{i=0}^{N-1} \bar{X}_{j,i} \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \quad (8-2)$$

1. (a,b) denotes the gcd of a and b

correspond to one of the 1-D DCT's of the data sequence $\{\hat{X}_{j,i}\}$ or equal to zero, with respect to m and n . That is, by defining

$$h_{j,l} = \sum_{i=0}^{N-1} \bar{X}_{j,i} \cos \frac{(4i+1)l}{2N} \pi, \quad (9)$$

we can see that eqn. (8-1) and eqn. (8-2) correspond to one of the $+h_{j,l}$ and $-h_{j,l}$ for some $l = 0, 1, \dots, N-1$, or equal to zero. Besides, through an index permutation, eqn. (9) can be implemented by a 1-D DCT as follow:

$$h_{j,l} = \sum_{i=0}^{N-1} \hat{X}_{j,i} \cos \frac{(2i+1)l}{2N} \pi, \quad (10-1)$$

where

$$\hat{X}_{j,i} = \begin{cases} \bar{X}_{j,i/2} & i:\text{even} \\ \bar{X}_{j,N-1-(i-1)/2} & i:\text{odd} \end{cases} \quad (10-2)$$

Hence, for the computation of an $(N \times N)$ -point 2-D DCT, only the computation of N N -point 1-D DCT's and some post-additions are required. Next, we will show that the post-addition stage can be implemented by a butterfly-like structure.

Since

$$\cos \frac{(4i+1)(m+(4(j+N/2)+1)n)}{2N} \pi = \pm \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi, \quad (11-1)$$

and

$$\cos \frac{(4i+1)(m-(4(j+N/2)+1)n)}{2N} \pi = \pm \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi. \quad (11-2)$$

$Y_{m,n}$ can be expressed as

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{j=0}^{N/2-1} \sum_{i=0}^{N-1} \left\{ \left(\bar{X}_{j,i} + \bar{X}_{(j+N/2),i} \right) \begin{pmatrix} \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi + \\ \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \end{pmatrix} \right\} & \text{for } n \text{ is even} \\ \frac{1}{2} \sum_{j=0}^{N/2-1} \sum_{i=0}^{N-1} \left\{ \left(\bar{X}_{j,i} - \bar{X}_{(j+N/2),i} \right) \begin{pmatrix} \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi + \\ \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \end{pmatrix} \right\} & \text{for } n \text{ is odd} \end{cases} \quad (12)$$

Let $\bar{X}_{j,i} + \bar{X}_{(j+N/2),i} = \bar{R}_{j,i}$,

$$\bar{X}_{j,i} - \bar{X}_{(j+N/2),i} = \bar{S}_{j,i}.$$

For n is even, since

$$\cos \frac{(4i+1)(m+(4(j+N/4)+1)n)}{2N} \pi = \pm \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi \quad (13-1)$$

and

$$\cos \frac{(4i+1)(m-(4(j+N/4)+1)n)}{2N} \pi = \pm \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi. \quad (13-2)$$

$Y_{m,n}$ can be written as

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{j=0}^{N/4-1} \sum_{i=0}^{N-1} \left\{ \begin{pmatrix} \left(\bar{R}_{j,i} + \bar{R}_{(j+N/4),i} \right) \times \\ \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi + \\ \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \end{pmatrix} \right\} & \text{for } n = 2(2k) \\ \frac{1}{2} \sum_{j=0}^{N/4-1} \sum_{i=0}^{N-1} \left\{ \begin{pmatrix} \left(\bar{R}_{j,i} - \bar{R}_{(j+N/4),i} \right) \times \\ \cos \frac{(4i+1)(m+(4j+1)n)}{2N} \pi + \\ \cos \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \end{pmatrix} \right\} & \text{for } n = 2(2k+1) \end{cases} \quad (14)$$

where $k = 0, 1, \dots, N/4-1$.

For example, if $N=4$, by eqn. (14) we have

$$Y_{m,2} = \frac{1}{2} \sum_{j=0}^3 \left\{ \left(\bar{R}_{0,j} - \bar{R}_{1,j} \right) \begin{pmatrix} \cos \frac{(4i+1)(m+2)}{2N} \pi + \\ \cos \frac{(4i+1)(m-2)}{2N} \pi \end{pmatrix} \right\}. \quad (15)$$

For n is odd, let

$$G_{j,l} = \sum_{i=0}^{N-1} \bar{S}_{j,i} \cos \frac{(4i+1)l}{2N} \pi \quad (16-1)$$

and

$$H_{j,l} = \sum_{i=0}^{N-1} \bar{S}_{j,i} \cos \frac{(4i+1)l}{2N} \pi. \quad (16-2)$$

Since

$$\begin{aligned} & \cos \frac{(4i+1)(m+(4(j+N/4)+1)n)}{2N} \pi \\ &= \mp \sin \frac{(4i+1)(m+(4j+1)n)}{2N} \pi \end{aligned}, \quad (17-1)$$

and

$$\begin{aligned} & \cos \frac{(4i+1)(m-(4(j+N/4)+1)n)}{2N} \pi \\ & = \pm \sin \frac{(4i+1)(m-(4j+1)n)}{2N} \pi \end{aligned} \quad (17-2)$$

But

$$\sin \frac{(4i+1)l}{2N} \pi = \cos \frac{(4i+1)(N-l)}{2N} \pi, \quad (18)$$

i.e. the 1-D discrete sine transform can be directly computed from the 1-D discrete cosine transform. Therefore, for some r and s, $0 \leq r, s \leq N$, $Y_{m,n}$ can be written as

$$Y_{m,n} = \begin{cases} \frac{1}{2} \sum_{j=0}^{N/4-1} \left\{ \left(\bar{G}_{jr} - \bar{G}_{(j+N/4)(N-r)} \right) + \left(\bar{H}_{js} + \bar{H}_{(j+N/4)(N-s)} \right) \right\} & \text{for } n = 2(2k) \\ \frac{1}{2} \sum_{j=0}^{N/4-1} \left\{ \left(\bar{G}_{jr} + \bar{G}_{(j+N/4)(N-r)} \right) + \left(\bar{H}_{js} - \bar{H}_{(j+N/4)(N-s)} \right) \right\} & \text{for } n = 2(2k+1) \end{cases} \quad (19)$$

where $k = 0, 1, \dots, N/4-1$, and \bar{G}_{jr} and \bar{H}_{js} are respectively equal to $\pm G_{jr}$ and $\pm H_{js}$. For example, if $N=4$, $r=3$ and $s=1$, by eqn. (19), we have

$$Y_{2,3} = \frac{1}{2} \left\{ (-G_{03} + G_{11}) + (H_{01} + H_{13}) \right\}.$$

As a result, the computation of an $(N \times N)$ -point 2-D DCT can be achieved by recursively applying the above decompositions (eqn. (14) and eqn. (19)). The signal flow graphs for a 4×4 and an 8×8 DCT's are shown in Figure 1 and Figure 2, respectively.

III. Complexity analysis of the post-addition stage

For an $(N \times N)$ -point 2-D DCT, let $A(N)$ and $B(N)$ respectively denote the number of additions required in the first $\log_2 N$ butterfly stages and the last butterfly stage, and let $C(N)$ denotes the number of nodes that don't require butterfly computations in the first $\log_2 N$ butterfly stages. From eqns. (15) and (18), we have $C(4) = 2$ and $C(N) = C(N/2) + N/2$ for $N \geq 8$, and $B(N) = N^2 -$

$2N$. Therefore, $A(N) = N^2 \log_2 N - C(N) + B(N) = N^2(1 + \log_2 N) - 3N + 2$.

IV. Conclusion

A new index-permutation based 2-D DCT algorithm has been presented in this paper. The succinct derivation of the proposed algorithm make it more easy to describe the processes of : how to map one 2-D DCT into a number of 1-D DCT's. Moreover, the structure of the post-addition stage of the proposed algorithm is more regular than that of [7], and a systematic approach for constructing the post-addition stage has also been described.

References

- [1] R. K. Rao and P. Yip, Discrete Cosine Transform: Algorithm, Advantages, and Applications. New York: Academic, 1990.
- [2] M. J. Narasimha and A. M. Peterson, "On the Computation of the Discrete Cosine Transform," IEEE Trans. Communications, Vol. COM-26, No. 6, pp. 934-936, June 1978.
- [3] H. S. Hou, "A Fast Recursive Algorithms for Computing the Discrete Cosine Transform," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-35, pp. 1455-1461, Oct. 1987.
- [4] PeiZong Lee and Fang-Yu Huang, "Restructured Recursive DCT and DST Algorithms," IEEE Trans. Signal Processing, vol. 42, no. 7, pp. 1600-1609, Jul. 1994.
- [5] P. Duhamel and C. Guillemot, "Polynomial Transform Computation of 2-D DCT," in proc. ICASSP '90, pp. 1515-1518, Apr. 1990.
- [6] E. Feig and S. Winograd, "Fast Algorithms for the Discrete Cosine

- Transform," IEEE Trans. Signal Processing, vol. 40, no. 9, pp. 2174-2193, Sept. 1992.
- [7] N. I. Cho and S.U. Lee, "Fast Algorithm and Implementation of 2-D DCT," IEEE Trans. Circuits Syst. vol. 38, no. 3, pp. 297-305, Mar. 1991.
- [8] N. I. Cho and S.U. Lee, "A Fast 4×4 DCT Algorithm for the Recursive 2-D DCT," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-40, no. 9, pp. 2166-2173, Sep. 1992.
- [9] N. I. Cho, I. D. Yun, and S.U. Lee, "On the Regular Structure for the Fast 2-D DCT Algorithm," IEEE Trans. Circuits Syst. vol. 40, no. 4, pp. 259-266, Apr. 1993.

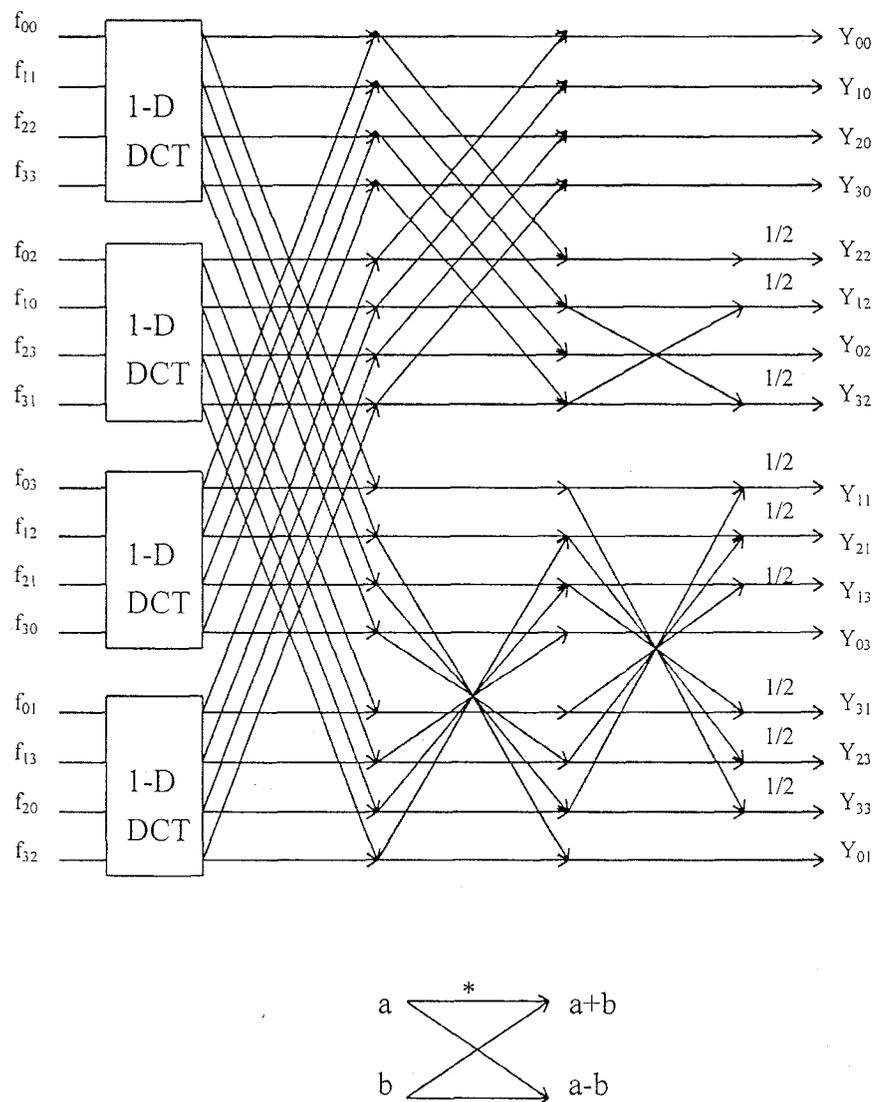
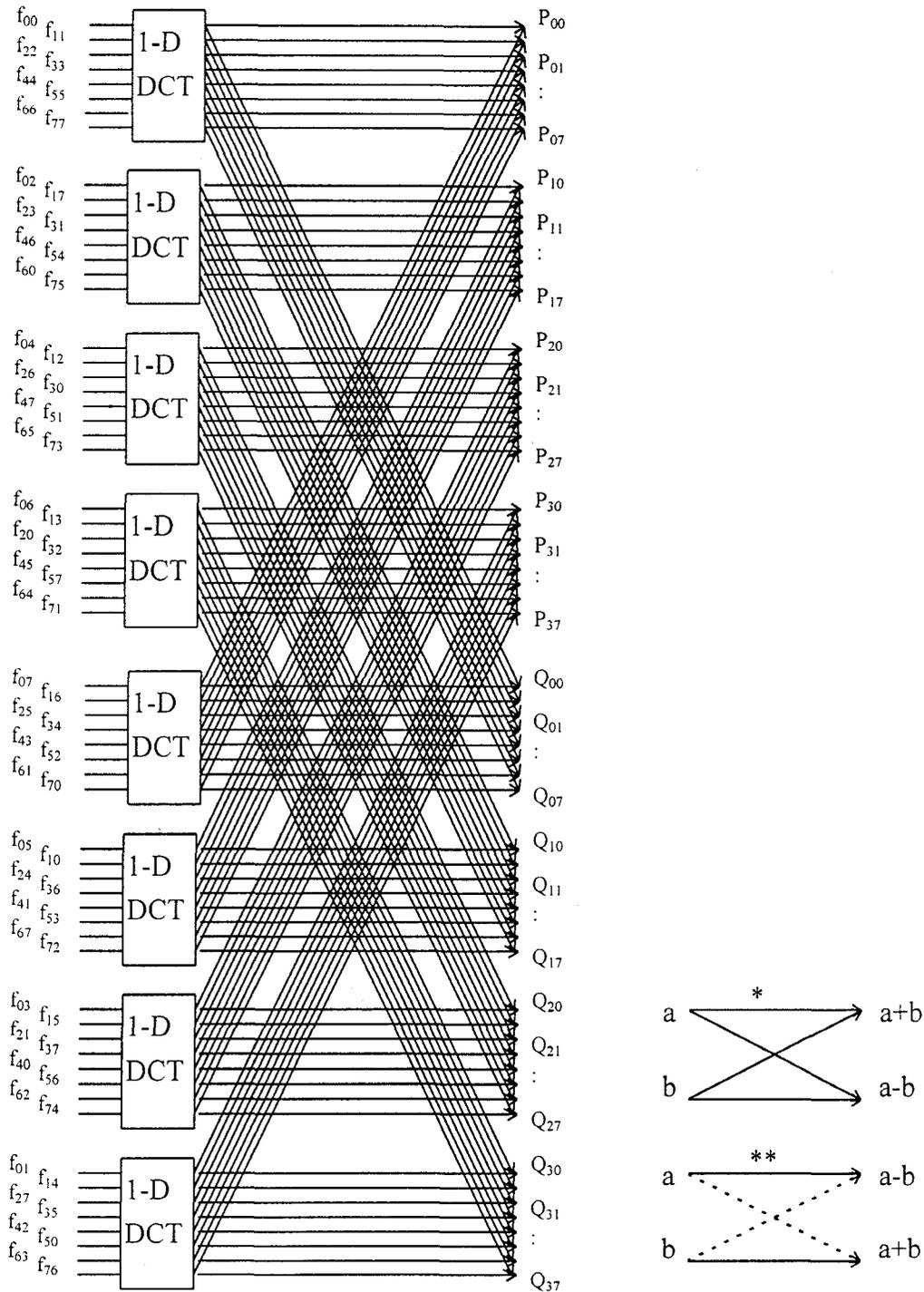


Figure 1. The signal flow graph for 4×4 DCT

(The solid butterfly * is of the normal butterfly structure as shown in [1].)



(a)

Figure 2. The signal flow graph for 8x8 DCT

(a) the 1st butterfly stage of the post-addition stage

(b) the 2nd, the 3th, and the 4th butterfly stages of the post-addition stage (for n is even)

(c) the 2nd, the 3th, and the 4th butterfly stages of the post-addition stage (for n is odd)

(The output order of the broken butterfly ** is reverse to that of the solid butterfly *.)

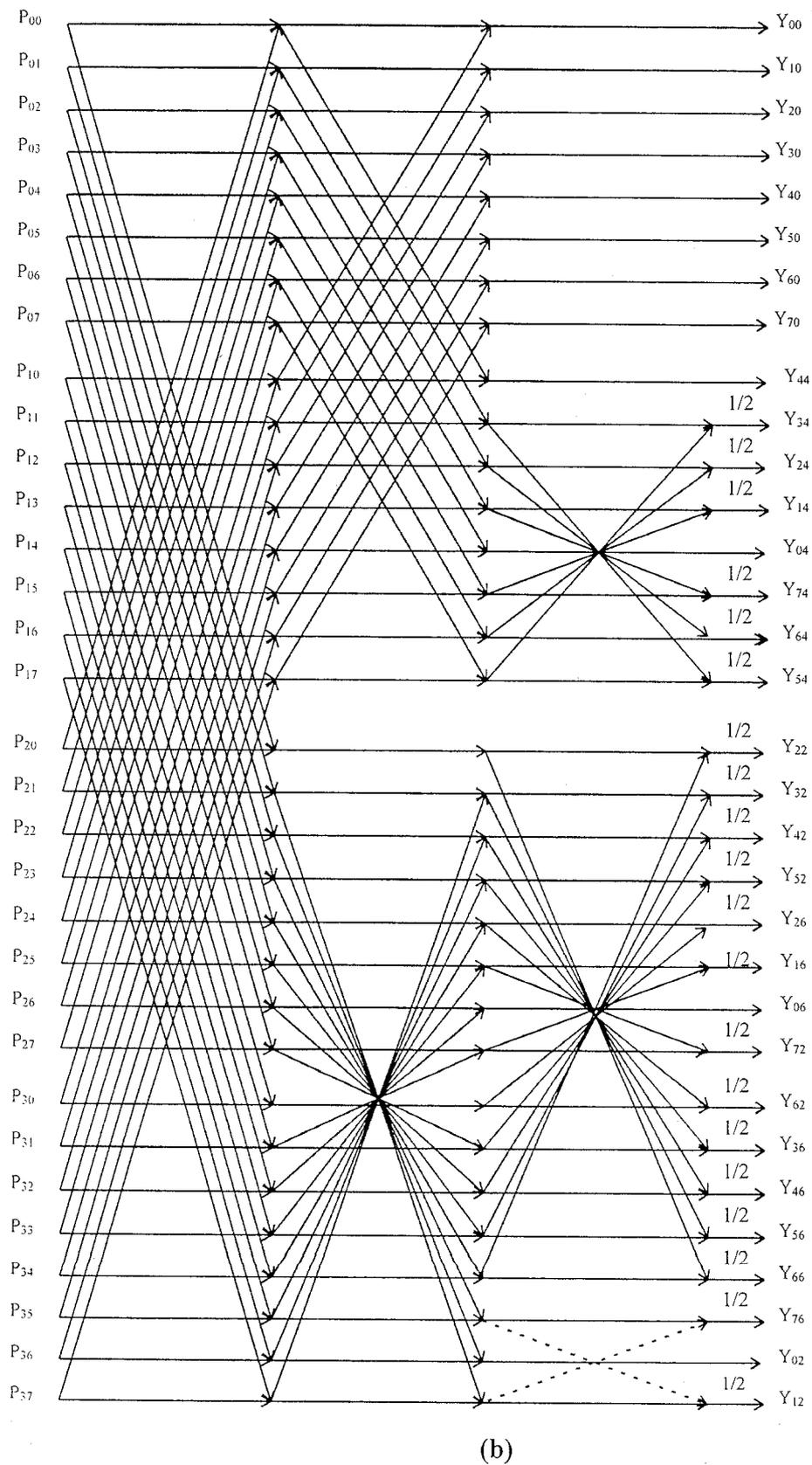
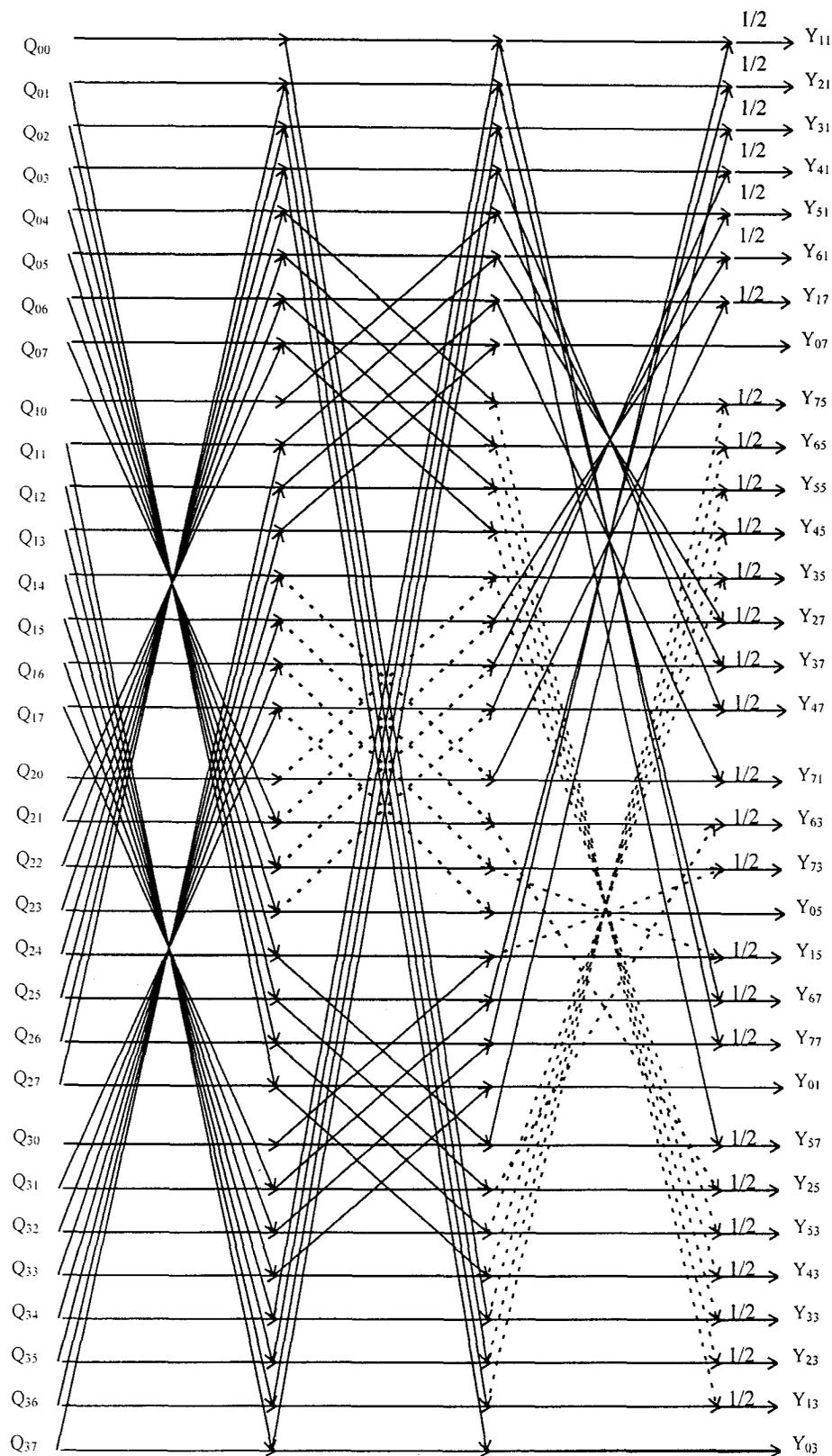


Figure 2. (Continued)



(c)

Figure 2. (Continued)