



Efficient Hardware Realization of 16x16 Discrete Cosine Transform

Herb-Dee Wu* and Chau-Yun Hsu**

*M200, Bldg. 11, 195 Sec. 4, Chung Hsing Rd., Chutung, Hsinchu, Taiwan 31015, R. O. C.

**40 Chung-Shan North Road, 3rd sec., Taipei, Taiwan, 10451, R.O.C.

RÉSUMÉ

Résumé—La transformation discrète de cosinus à deux dimensions (2-D DCT) a été largement appliquée dans la compression des signaux d'images et de vidéo. L'emploi direct dans le hardware du 16x16 DCT exige plusieurs multiplicateurs. Puisque le circuit de multiplicateurs est très complexe, nous proposons un hardware efficace réalisant le 16x16 DCT. Nous présentons ici une architecture exigeant aucun multiplicateur. Une architecture concomitante pour un 16x16 DCT qui emploie l'algorithme de coefficient de différence permuté est aussi présenté. Les avantages de cette nouvelle architecture sont les suivants: grande rapidité, haute précision, et hardware efficace. Son efficacité pour 16x16 DCT a été démontrée en se servant de la simulation de longueurs démotés limitées. Les résultats montrent que notre technique nouvelle est très utile pour la compression d'images numériques.

ABSTRACT

Abstract—The two-dimensional discrete cosine transform (2-D DCT) has been widely applied in image and video signal compression. A direct hardware implementation of the 16x16 DCT requires many multipliers. Due to the circuit of a multiplier is very complex, we propose an efficient hardware realization of 16x16 DCT. The new structure requires no multipliers. A concurrent architecture for a 16x16 DCT by using the permuted difference coefficient (PDC) algorithm is presented in this paper. The advantages of this new architecture are: high speed; high accuracy; and efficient hardware realization. This proposed architecture has shown its performance for 16x16 DCT with the simulation of finite word-length. The simulation results shows that our development is quite attractive for digital image compression.

I. INTRODUCTION

The discrete cosine transform (DCT), first proposed by Ahmed et al. in 1974, is famous for its performance closer to the optimal KLT for highly correlated input data [1]. The two-dimensional discrete cosine transform (2-D DCT) has been widely applied in image and video signal compression such as videophone, teleconferencing, and HDTV.

The 2-D DCT is defined as

$$Z = C^t X C, \quad (1)$$

where

X : data matrix of size $N \times N$,
C : normalized cosine coefficients (normalized transform kernels) matrix of size $N \times N$, the (k, l) th element of matrix C is

$$c_{k,l} = \frac{\sqrt{2}}{\sqrt{N}} \cos\left(\frac{\pi(2k-1)(l-1)}{2N}\right), \quad (2a)$$

$$\text{for } k = 1, 2, \dots, N, \\ l = 2, 3, \dots, N,$$

$$c_{k,l} = \frac{1}{\sqrt{N}}, \quad (2b)$$

$$\text{for } k = 1, 2, \dots, N, \\ l = 1,$$

Z : transform-domain data matrix of size $N \times N$,

C^t : the transpose of matrix C.

A direct implementation of eq. (1) requires a large amount of computation. Many algorithms have been developed to compute the DCT by using variously forms of butterfly structures with few number of multipliers [2]-[4]. However, many multipliers are still required for high speed realization of the 2-D DCT.

The proposed architecture requires no multipliers. In this research, we develop a concurrent architecture by using pipeline structure, parallel structure, and permuted difference

coefficient (PDC) algorithm [5]-[6] to achieve a high accuracy, high speed, and efficient hardware realization of the 2-D DCT.

II. PERMUTED DIFFERENCE COEFFICIENT ALGORITHM

The permuted difference coefficient (PDC) technique has been used for the implementation of the FIR digital filter and that of the discrete Fourier transform (DFT) [5]-[6]. The PDC method is modified by us, so that it is suitable to compute the DCT.

The modified PDC algorithm can be described as follows:

step 1: Reorder the original coefficients in a sequence with rising magnitude.

step 2: Difference coefficients are obtained by the difference of successive values for the reordered coefficients.

step 3: As step 1 and step 2 are repeatedly performed, the high order difference coefficients can be also obtained. The process is repeated, until the all of the difference coefficients are equal to the integer powers of two or zero.

Let input data be x_1, x_2, \dots, x_N and coefficients be c_1, c_2, \dots, c_N . The output, y , is given as

$$y = \sum_{m=1}^N x_m c_m. \quad (3)$$

Let c_i^* be the absolute value of c_m , where i is the reordered index. Then, c_i^* is derived by the following conditions



$$(i) c_i^* = |c_m|, \quad (4a)$$

$$(ii) 0 \leq c_1^* \leq c_2^* \leq \dots \leq c_N^*. \quad (4b)$$

Consequently eq. (3) can be rewritten by

$$y = \sum_{i=1}^N x_i^* c_i^*, \quad (5)$$

$$\text{where } x_i^* = \text{sign}(c_m) \cdot x_m. \quad (6)$$

First-order permuted difference coefficients $d_i^{(1)}$ are defined as follows:

$$d_1^{(1)} = c_1^*, \quad (7a)$$

$$d_i^{(1)} = c_i^* - c_{i-1}^*, \quad \text{for } i = 2, 3, \dots, N. \quad (7b)$$

Using $d_i^{(1)}$, eq. (5) is rewritten by

$$y = \sum_{i=1}^N s_i^{(1)} d_i^{(1)}, \quad (8)$$

where

$$s_i^{(1)} = \sum_{n=i}^N x_n^*, \quad (9)$$

As a consequence, y defined in eq. (3) can be calculated by using eq. (8).

High-order permuted difference coefficients are obtained similarly as first-order permuted difference coefficients.

III THE PROPOSED ARCHITECTURE

Because of the separability of the 2-D DCT cosine coefficients, the 2-D 16x16 DCT can be implemented by two 1-D 16x1 DCT. A block diagram of the 16x16 DCT is shown in Fig. 1 [7]. The two 1-D DCT operate concurrently.

Let the 1-D DCT of an $N \times N$ data matrix X be presented as $Y=XC$. The (k, l) th element of Y is given by

$$y_{k,l} = \sum_{m=1}^N x_{k,m} c_{m,l}, \quad (10)$$

$y_{k,l}$ can be computed by using permuted difference coefficient (PDC) algorithm [5]-[6] for each column of the normalized cosine coefficients matrix C . Consequently for each k th row vector $(x_{k,1}, x_{k,2}, \dots, x_{k,N})$ of X , a row vector $(y_{k,1}, y_{k,2}, \dots, y_{k,N})$ can be computed concurrently by PDC algorithm.

From the properties of the matrix C , eq. (10) can be rewritten by

$$y_{k,l} = \sum_{m=1}^{N/2} u_{k,m} c_{m,l}, \quad \text{for } l = 1, 3, \dots, N-1, \quad (11a)$$

$$\text{where } u_{k,m} = x_{k,m} + x_{k,N-m+1},$$

and

$$y_{k,l} = \sum_{m=1}^{N/2} v_{k,m} c_{m,l}, \quad \text{for } l = 2, 4, \dots, N, \quad (11b)$$

$$\text{where } v_{k,m} = x_{k,m} - x_{k,N-m+1}.$$

Fig. 2 shows the block diagram of the 16x1 DCT.

In Fig. 2, the input sequence $x_{k,1}, x_{k,2}, \dots, x_{k,16}$ is bit-parallel shifted sequentially into the D shift registers at every clock time interval (T). After every 16 clock time interval (16T), the contents of the D shift registers are bit-parallel loaded into the L latch registers simultaneously. From Fig. 2, it is easy to see that 16 PDC's work simultaneously so that the 16x1 DCT can operate with high speed. The output data $y_{k,1}, y_{k,2}, \dots, y_{k,16}$ are computed by PDC(1), PDC(2), ..., PDC(16), respectively. Every 16 clock time interval (16T), the output data $y_{k,1}, y_{k,2}, \dots, y_{k,16}$ are also concurrently bit-parallel loaded into the U shift registers. Then, the contents of the U shift registers are bit-parallel shifted out sequentially at every clock time interval (T). Because each output data, y , is generated at every clock time interval (T), our architecture achieves a high throughput rate. The D shift registers, 16 PDC's, and U shift registers constitute a pipeline structure. They operate concurrently.

There are 16 PDC's in Fig. 2. They are similar for each other. The PDC(6) is used to illustrate the PDC algorithm. Table 1 shows the original coefficients, the first-order and second-order permuted difference coefficients in PDC(6). The hardware realization of PDC(6) is illustrated in Fig. 3.

The circuit, which is shown in Fig. 3, is composed of adders only.

IV. SIMULATION RESULTS

The proposed architecture of the 2-D DCT adopts the fixed-point arithmetic. The block diagram for testing the accuracy with finite word-length is shown in Fig. 4 [7]. The original input of the 2-D DCT is in 9-bit 2's complement form. Let the output of the first 1-D DCT, the output of the second 1-D DCT, and the cosine coefficients word-length is n_1 , n_2 , and n bits, respectively. The three parameters (n_1 , n_2 , and n) can be determined by simulation.

The PSNR is defined by

$$\text{PSNR} = 20 \log \frac{255}{\text{root mean square error}} \text{ (dB)}. \quad (12)$$

Four test pictures "usc-girl" ("Lenna"), "pepper", "baboon", and "first-order Markov signal with $\rho=0.95$ " are used to calculate the PSNR for various n_1 , n_2 , and n .

Fig. 5 shows the resultant PSNR with n_1 and n_2 as parameters for test picture "usc-girl" ("Lenna"). The word-length for the cosine coefficients is set to be 14 bits.

Fig. 6 shows the resultant PSNR with various n (cosine coefficients word-length) for test pictures "usc-girl".

The simulation results, with four test pictures, are similar for each other.

From the simulation results, we can easily determine the reasonable values of the n_1 , n_2 , and n for the proposed fixed-point architecture. These reasonable values are: $n_1=12$; $n_2=14$; and $n=7$.

The reasonable values of the n_1 , n_2 , and n , shown above, are used in our architecture. The PSNR is above 54dB for each of the four test pictures. No visible distortion has been observed under real images testing.

V. CONCLUSIONS

In this paper, a concurrent architecture for implementing a 16x16 DCT is presented. We may conclude the merits of this proposed architecture as following: (1) It is an efficient architecture because it requires no multipliers and the major parts of the circuits are adders. (2) There is no need for the ROM tables to store the cosine

coefficients (transform kernels). (3) This architecture can be extended easily to arbitrary data length of DCT. For other algorithms, in general, the data length, N , must be equal to the integer powers of two. (4) The higher accuracy it perform because it has fewer rounding or truncation stages than that of other algorithms. (5) The high speed are achieved via using pipeline structure, parallel structure, and the replacement of multipliers by adders.

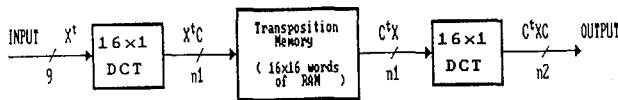


Fig. 1 Block diagram of the 16x16 DCT.

Table 1
Original and Permuted Difference Coefficients in PDC(6) for 16x1 DCT

m	c_m	i	c_i^*	$d_i^{(1)}$	j	$d_j^{(1)*}$	$d_j^{(2)}$	p	$d_p^{(2)*}$
1	40	1	4	4	1	2	2	1	0
2	4	2	13	9	2	3	1	2	1
3	-35	3	21	8	3	4	1	3	1
4	-43	4	29	8	4	5	1	4	1
5	-13	5	35	6	5	6	1	5	1
6	29	6	40	5	6	8	2	6	1
7	45	7	43	3	7	8	0	7	2
8	21	8	45	2	8	9	1	8	2

Note: The above coefficient values must be multiplied by a scale factor 2^{-7} .

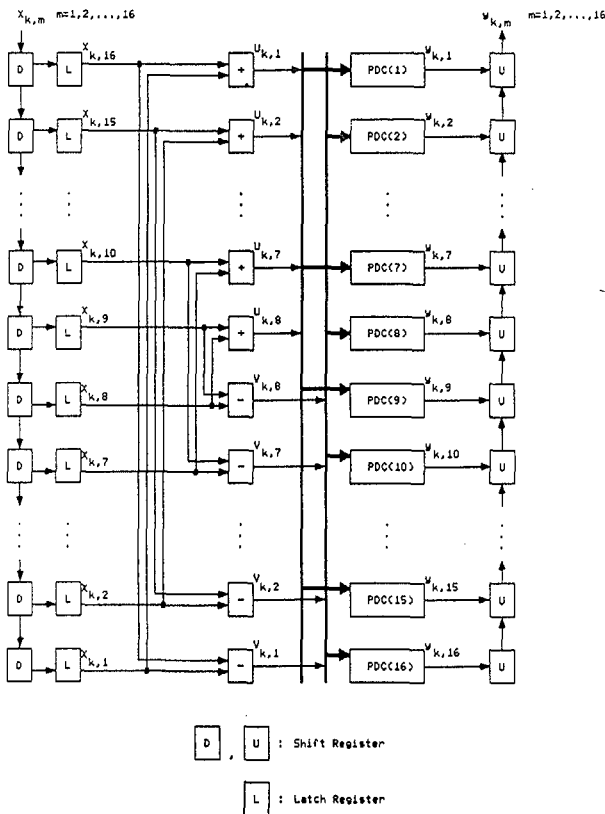


Fig. 2 Block diagram of the 16x1 DCT.

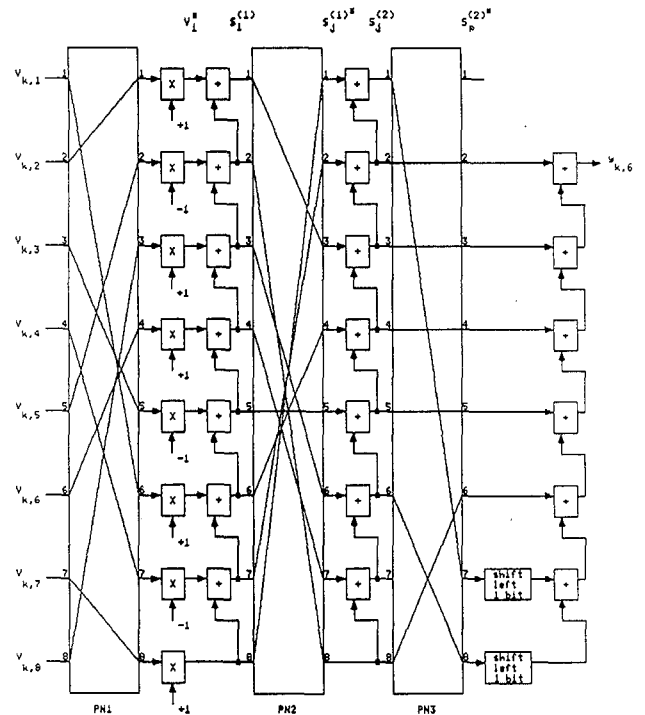


Fig. 3 Hardware realization for PDC(6) which is correspond to Table 1.

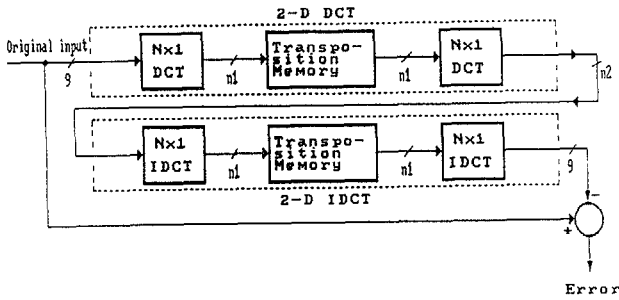


Fig. 4 The block diagram for testing the accuracy.

REFERENCES

- [1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90-93, Jan. 1974.
- [2] W. H. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithms for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, Sept. 1977.
- [3] B. G. Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 1243-1245, Dec. 1984.
- [4] 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.
- [5] K. Nakayama, "Permuted difference coefficient realization of FIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 269-278, Apr. 1982.
- [6] G. F. Boudreaux-Bartels and T. W. Parks, "Discrete Fourier transform using summation by parts," *International Conference on Acoust., Speech, Signal Processing*, Dallas, Texas, pp. 1827-1830, Apr. 1987.
- [7] M. T. Sun, T. C. Chen, and A. Gottlieb, "VLSI implementation of a 16x16 discrete cosine transform," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 610-617, Apr. 1989.

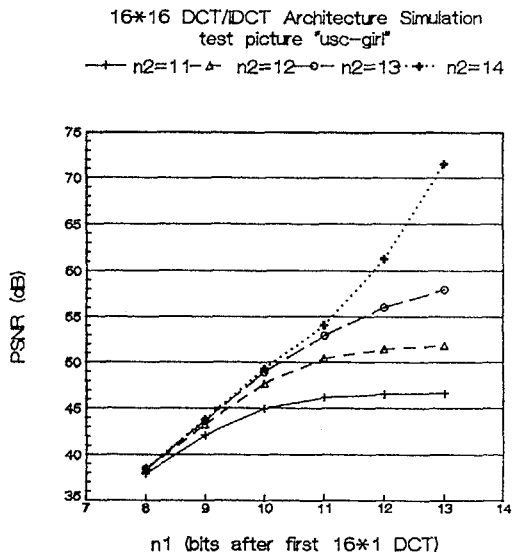


Fig. 5 Simulation results of the proposed fixed-point architecture for various n1 and n2. The n is set to be 14 bits.

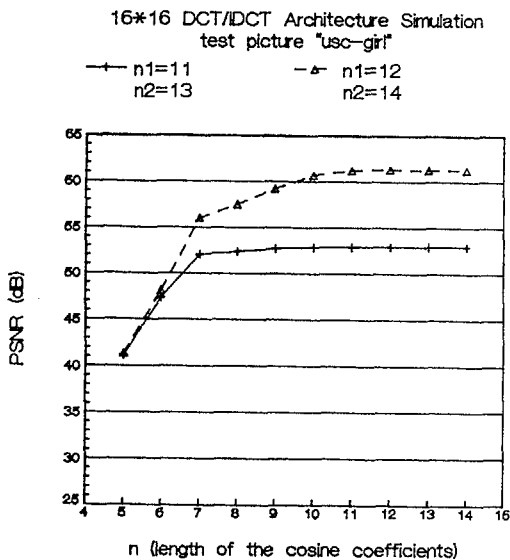


Fig. 6 Simulation results of the proposed fixed-point architecture for various n.