EFFICIENT VLSI IMPLEMENTATION OF INVERSE DISCRETE COSINE TRANSFORM

J. Lee, N. Vijaykrishnan, M. J. Irwin

Embedded & Mobile Computing Design Center
The Pennsylvania State University
E-mail: {joolee, vijay, mji}@cse.psu.edu

ABSTRACT

In this paper, a novel 2-D IDCT architecture based on the energy compaction property of 2-D DCT is proposed. This architecture performs 2-D IDCT directly on the 2-D DCT data set, avoiding the need for the transposition memory. We derive a recursion equation from the definition of the 2-D IDCT algorithm and use it to implement a wavefront array processor. The wavefront array processor consists of highly regular, parallel and pipelined processing elements which are suitable for VLSI implementation. This implementation also utilizes the sparseness property of the 2-D DCT coefficients to reduce the computational complexity. It is shown that the proposed architecture achieves a high throughput rate, $(15+m)$ clock cycles per 2-D DCT data set, where m is the number of the non-zero DCT coefficients. Another important aspect of this architecture is that it provides an efficient way to control the trade-off between visual quality of the reconstructed image and computational complexity.

1. INTRODUCTION

Due to the decorrelation and energy compaction properties of DCT [1] for typical image and video data, most of the current image/video standards, such as JPEG, H.26x and MPEG family, use discrete cosine transform (DCT) to remove spatial redundancies. DCT and inverse DCT (IDCT) are computationally intensive algorithms. Its direct computation of 2-D $N \times N$ DCT (IDCT) requires $O(N^4)$ multiplications, which need efficient VLSI architecture to meet the constraints of various real-time applications.

Many researchers have proposed a large number of efficient DCT/IDCT architectures, such as fast algorithm-based designs, multiplier-based designs, adder-based designs, memory-based designs and so on. Most of the above design approaches use a row-column decomposition method [2]. 2-D DCT (or 2-D IDCT) is decomposed into two separate 1-D DCT's (or 1-D IDCT's). That is, the row (or column) data are processed using 1-D DCT (or 1-D IDCT) first and results are stored in the transposition memory. Then, its column (or row) data are processed again using 1-D DCT (or 1-D IDCT), which yields 2-D DCT (or 2-D IDCT) results. This row-column decomposition scheme is shown in Fig. 1. Since many existing 1-D DCT (or 1-D IDCT) techniques can be applied directly, this approach has been widely used. But its disadvantage is that the fast transposition circuit, in general, occupies large area of the chip [3]. There are other approaches which perform 2-D DCT (or 2-D IDCT) directly on the 2-D data set with less number of multiplications [4, 5]. These algorithms often need direct mapping of complex signal flow graphs to the corresponding hardware components, which increases implementation complexity of VLSI circuits.

In this paper, we propose a novel 2-D IDCT architecture based on the energy compaction property of 2-D DCT. The energy compaction property has also been shown to be useful in designing an architecture for the motion estimation algorithm in [6]. The focus of this work is on utilizing this property for 2-D IDCT implementation and comparing its performance with other 2-D IDCT architectures. The rest of this paper is organized as follows. In Sections 2 and 3, the decorrelation and energy compaction properties of 2-D DCT are briefly explained and a recursion equation from the 2-D IDCT definition is derived. Further, we present a novel architecture for 2-D IDCT. In Section 4, we show that the proposed 2-D IDCT architecture can achieve a higher throughput rate with low initial delay as compared with other 2-D IDCT architectures. In Section 5, we briefly summarize our work and conclude the paper.

2. PROPOSED ALGORITHM

For a given 2-D spatial data sequence $x(i,j)$, $0 \le i,j \le N-1$, the corresponding 2-D DCT data sequence $X(u,v)$, $0 \le u,v \le N-1$, is defined as

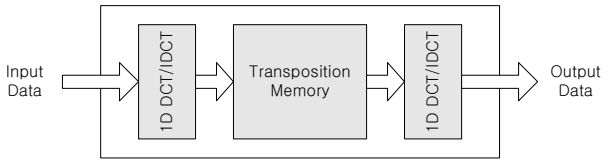


Fig. 1. Row-column decomposition approach

$$X(u,v) = \frac{2}{N} C(u)C(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i,j) \cos \frac{(2i+1)u\pi}{2N} \times \cos \frac{(2j+1)v\pi}{2N} \quad (1)$$

where

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & 1 \le k \le N-1 \end{cases} \quad (2)$$

And, its 2-D IDCT is defined as

$$x(i,j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)X(u,v) \cos \frac{(2i+1)u\pi}{2N} \times \cos \frac{(2j+1)v\pi}{2N} \quad (3)$$

Therefore, $N \times N$ DCT matrix $T = \{t(m,n)\}$, where $t(m,n)$ represents the matrix entry in the m_{th} row and the n_{th} column, is given by

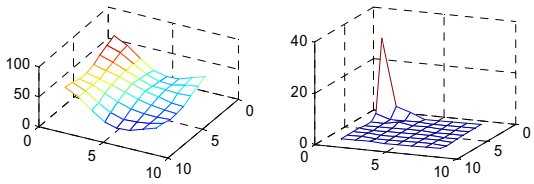
$$t(m,n) = \begin{cases} \frac{1}{\sqrt{N}}, & m = 0 \\ & 0 \le n \le N-1 \\ \sqrt{\frac{2}{N}} \cos \frac{(2n+1)m\pi}{2N}, & 1 \le m \le N-1 \\ & 0 \le n \le N-1 \end{cases} \quad (4)$$

The 2-D DCT and 2-D IDCT can be written in a matrix form as

$$X = TxT^t \quad (5)$$

$$x = T^t XT \quad (6)$$

The case of $N = 8$ is considered, since image and video compression standards use 8×8 block for 2-D DCT and 2-D IDCT operations. Since the basis vectors of the DCT are orthogonal, IDCT can be easily obtained, as shown in eq. (6). Due to the similarity between eq. (5) and eq. (6), the previous architectures developed for DCT by many researchers, have been also used for IDCT operations. But 2-D spatial data matrix x and 2-D DCT data matrix X have different signal characteristics. DCT transforms highly correlated image into a few transform coefficients, as shown in Fig. 2. Moreover, the conventional image/video coding techniques use quantization process to achieve higher compression ratio. Therefore, the 2-D DCT data matrix X has a few non-zero coefficients in the low frequency zone, which makes it possible to design more efficient IDCT architecture than the previous DCT-based IDCT architectures [9].



(a) 2-D spatial data (b) 2-D DCT data

Fig. 2. Examples of 2-D spatial and 2-D DCT data

In this paper, a new 2-D IDCT architecture to fully utilize the energy compaction property and the sparseness property of the 2-D DCT data matrix, X is proposed. Let $X(m,n)$ be the matrix entry in the m_{th} row and the n_{th} column of the 2-D DCT data matrix X and the row vector $T(n)$ be the n_{th} row of the DCT matrix T. Therefore, the equation (6) can be represented as

$$x = T^t XT = \sum_{m=0}^{7} \sum_{n=0}^{7} X(m,n)T(m)^t T(n) \quad (7)$$

Equation (7) can be written in zig-zag scan order as follows,

$$\begin{aligned} x &= \sum_{m=0}^{7} \sum_{n=0}^{7} X(m,n)T(m)^t T(n) \\ &= X(0,0)T(0)^t T(0) + X(0,1)T(0)^t T(1) \\ &+ X(1,0)T(1)^t T(0) + X(2,0)T(2)^t T(0) \\ &+ \ldots\ldots + X(7,6)T(7)^t T(6) + X(7,7)T(7)^t T(7) \end{aligned} \quad (8)$$

The 2-D spatial data matrix x in eq. (8) can be considered as linear combinations of basis images which are obtained by "outer product" of the column vector $T(m)^t$ and the row vector $T(n)$. This interpretation of eq. (8) can make it easier to manipulate the sparseness of the 2-D DCT data matrix X to calculate the 2-D spatial data matrix x, as explained in Section 3. It indicates that the trade-off between visual quality of the reconstructed image and computational complexity can be controlled by using only portions of the weighting factor matrix X, (i.e., different number of elements from $X(m,n)$). More elements are chosen for higher quality, while a smaller number is preferred for reducing the computational complexity.

3. IMPLEMENTATION OF THE PROPOSED ALGORITHM

2-D IDCT algorithm is implemented with wavefront array processing architecture as shown in Fig. 3. The values of IDCT kernels enter the top and the left sides of the proposed architecture. The data movement propagates from the top-left corner to the bottom-right corner of the processing elements. These computational wavefronts are

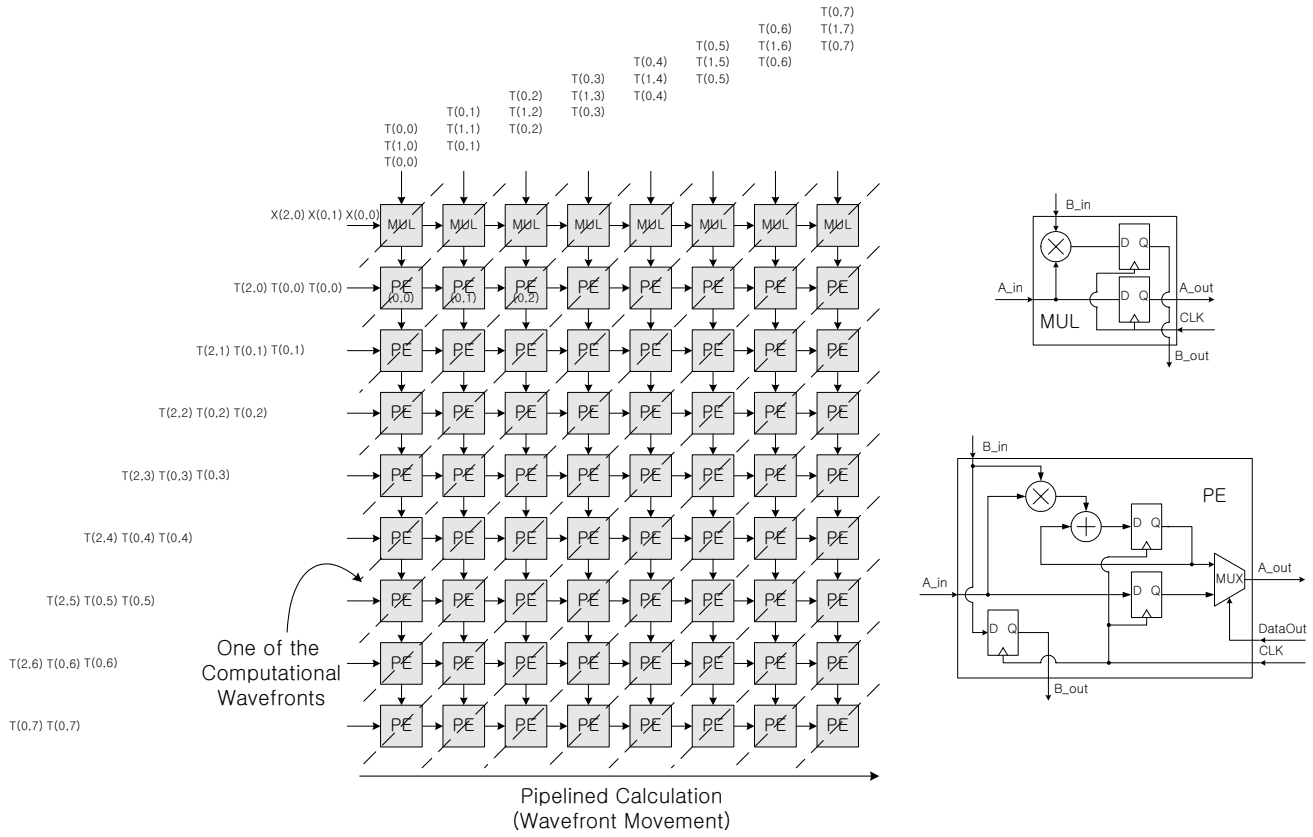


Fig. 3. Implementation of the proposed 2-D IDCT architecture

Table 1. Operations of PEs

	PE(0,0)	PE(0,1)	PE(0,2)	⇒
1	$x^1(0,0) = X(0,0)T(0,0)T(0,0)$			
2	$x^2(0,0) = x^1(0,0) + X(0,1)T(0,0)T(1,0)$	$x^1(0,1) = X(0,0)T(0,0)T(0,1)$		
3	$x^3(0,0) = x^2(0,0) + X(2,0)T(2,0)T(0,0)$	$x^2(0,1) = x^1(0,1) + X(0,1)T(0,0)T(1,1)$	$x^1(0,2) = X(0,0)T(0,0)T(0,2)$	⇓
4		$x^3(0,1) = x^2(0,1) + X(2,0)T(2,0)T(0,1)$	$x^2(0,2) = x^1(0,2) + X(0,1)T(0,0)T(1,2)$	
5			$x^3(0,2) = x^2(0,2) + X(2,0)T(2,0)T(0,2)$	

pipelined on the processing elements array to achieve a high throughput rate. In the proposed architecture, one wavefront corresponds to one mathematical recursion, i.e., one "outer product" of the column vector $T(m)^t$ and the row vector $T(n)$, scaled by the non-zero DCT coefficient $X(m,n)$. Therefore, successive pipelining of the computational wavefronts through the processing elements will provide 2-D IDCT results from low visual quality of the image to high visual quality of the image, according to the number of the mathematical recursions as shown in eq. (9).

$$x^k = x^{k-1} + X(m,n)T(m)^t T(n) \tag{9}$$

For example, let us examine how 2-D IDCT is performed by fully utilizing the sparseness property of the 2-D DCT data matrix X. Let the 2-D DCT data matrix X have the following non-zero elements after quantization process.

$$X = \begin{bmatrix} X(0,0) & X(0,1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ X(2,0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

The data flow of the 2-D IDCT operations using this example is indicated in Fig. 3. Each pipelined wavefront

Table 2. Performance comparison of the different 2-D IDCT architectures

	Ma [7]	Lim [8]	Chang [9]	Proposed
Algorithmic Approach	Row-column decomposition	Row-column decomposition	Row-column decomposition	Direct 2-D algorithm
No. of Multipliers	$4N(N+1)$	$N \times N$	$N \times N$	$N(N+1)$
Transposition Memory	No	No	No	No
Total cycles per $N \times N$ 2-D IDCT	N^2	$2N$	N	m
I/O Ports	Serial In Parallel Out	Parallel In Parallel Out	Parallel In Parallel Out	Serial In Parallel Out

m : the number of the non-zero elements of the 2-D DCT matrix X

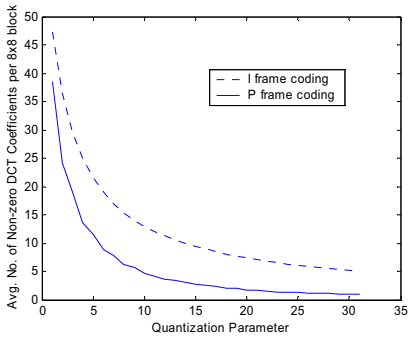


Fig. 4. Average number of non-zero DCT coefficients per 8×8 block vs. quantization parameter

performs one outer product, scaled by the non-zero DCT coefficient $X(m,n)$ and the recursion in eq. (9) gives the final results of the 2-D IDCT in eq. (8). Table 1 shows a representative set of PEs and the operations that they perform.

4. PERFORMANCE COMPARISON

For the experiment, H.263 video encoder is used. Fig. 4 shows the example of the average number of non-zero DCT coefficients per 8×8 block in the I and P frame coding mode. It becomes more apparent that fewer DCT coefficients survive after quantization process in the P frame coding. As the quantization parameter increases, the number of the non-zero DCT coefficients also decreases rapidly. Therefore, the proposed architecture can provide a high throughput rate with the pipelined processing elements, since only non-zero elements of the 2-D DCT data matrix X are used to perform 2-D IDCT. The total processing time required for 2-D IDCT is $(15+m)$ clock cycles, where 15 clock cycles are the initial delay and m is the number of the non-zero elements of the matrix X. The performance of the proposed 2-D IDCT architecture is compared with other 2-D IDCT architectures, as shown in Table 2. The proposed serial-in, parallel-out architecture gives high performance comparable to parallel-in, parallel-out architectures.

5. CONCLUSION

A novel architecture for 2-D IDCT is proposed. Pipelined array processor achieves a high throughput rate by fully utilizing the sparseness property of the 2-D DCT data matrix X. It does not require the transposition memory and consists of highly regular, parallel and pipelined processing elements which are suitable for VLSI implementation. Another important aspect is that this architecture can provide very efficient way to control the trade-off between visual quality of the reconstructed image and computational complexity by using different number of the non-zero DCT coefficients, $X(m,n)$.

6. REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computer*, Vol. C-23, pp. 90-93, Jan. 1974.

[2] K. Kim and J. Koh, "An area efficient DCT architecture for MPEG-2 video encoder," *IEEE Transactions on Consumer Electronics*, Vol. 45, pp. 62-67, Feb. 1999.

[3] P. A. Ruetz and P. Teng, "A 160-Mpixel/s IDCT processor for HDTV," *IEEE Micro*, Vol. 12, pp. 28-32, Oct. 1992.

[4] T. Chang, C. Kung, and C. Jen, "A simple processor core design for DCT/IDCT," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, pp. 439-447, April 2000.

[5] Y. Lee, T. Chen, L. Chen, M. Chen, and C. Ku, "A cost-effective architecture for 8×8 two-dimensional DCT/IDCT using direct method," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, pp. 459-467, June 1997.

[6] J. Lee, N. Vijaykrishnan, M. J. Irwin, and W. Wolf, "An Architecture for Motion Estimation in the Transform Domain," *International Conference on VLSI Design*, Jan. 2004.

[7] W. Ma, "2-D DCT systolic array implementation," *Electronics Letters*, Vol. 27, pp. 201-202, Jan. 1991.

[8] H. Lim and E. E. Swartzlander, Jr.; "A systolic array for 2-D DFT and 2-D DCT," *International Conference on Application Specific Array Processors*, pp. 123-131, Aug. 1994.

[9] Y. Chang and C. Wang, "New systolic array implementation of the 2-D discrete cosine transform and its inverse," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5, pp. 150-157, April 1995.