A New Systolic Array for Two-Dimensional Discrete Cosine Transform

هيكل انقباضي جديد للتحول الجيبي التمامي المتقطع ثنائي الأبعاد

Samia A. Mashali
Computer & Systems Dept., Electronics Research Institute
Cairo, Egypt.
E-mail : ERI @ FRCU.EUN.EG

Abstract -An efficient systolic implementation of the two-dimensional discrete cosine transform (2-D DCT) is developed. It offers a highly regular structure ideally suited for VLSI implementation. The computation complexity of the proposed implementation is O(N) as compared to the two-dimensional fast cosine transform algorithm of $O(N^2)$ [1].

ملخص البحث:

يعرض هذا البحث هيكل انقباضي للتحول الجيبي التمامي المتقطع ثنائي الابعاد ذو كفاءة عالية ويوفر هذا الهيكل تركيبة منتظمة تلائم تنفيذ الدوائر المتكاملة عديدة الاتساع.

وبالحساب يتبين ان درجة تعقيد الخوارزم تتناسب مع N وحده تعتبر افضل من خوارزم التحويل الجيبي التمامي السريع ثنائي الأبعاد والذي تصل درجة تعقيده الى N^2

I - Introduction

The discrete cosine transform (DCT) is a better approximation to the statistically optimal Karhunen - loeve transform (KLT) than most other orthogonal transforms [2],[3]. Therefore, it plays a very important role in image and speech coding.

The DCT transform requires massive and complicated data manipulation that lead to consideration and implementations employing parallelism and pipelining. The systolic array has the advantages of pipelinability, regularity, locality, and scalability, thus making it very suitable for VLSI signal processing [4]-[6].

Recently, Anew recursive algorithm for computing the 1-D DCT [7],[8] and the 2-D DCT [1] is introduced. The algorithm is similar to a decimation-in-frequency Cooley - Tukey FFT algorithm. Its computation complexity is $O(N^2)$. In this paper a new systolic array for the 2-D DCT is developed. It reduces the computation complexity to O(N).

The approach for mapping algorithms onto Systolic hardware can be

Similarly

$$C_{k_2}^{2n_2+1} = [(2C_{k_2}^{1})^2 - 2]\, C_{k_2}^{2n_2-1} - C_{k_2}^{2n_2-3} \qquad (10)$$

It is evident that we can derive one recurrence relation among coefficients of three successive data points when transforming for a fixed n. This leads to the introduction of systolic array schemes in hardware design.

III- Proposed Systolic Array

If we consider the DCT using (4),(9) & (10) simultaneously, the complete recurrence formulae are :

$$C_{k_1}^{2n_1+1} = A(C_{k_1}^{1})\, C_{k_1}^{2n_1-1} - C_{k_1}^{2n_1-3} \qquad (11)$$

$$C_{k_2}^{2n_2+1} = A(C_{k_2}^{1})\, C_{k_2}^{2n_2-1} - C_{k_2}^{2n_2-3} \qquad (12)$$

$$Z_{k1,k2}(n_1,n_2) = x(n_1,n_2)\, C_{k_1}^{2n_1+1} C_{k_2}^{2n_2+1} + Z_{k1,k2}(n_1-1,n_2) \qquad (13)$$

Where

&

$$A(C_{k_2}^{1}) = [(2\, C_{k_2}^{1})^2 - 2]$$

The systolic array for realizing the above recurrence formulae is shown in Fig 1. It requires (N_1N_2) processing elements. Four types of processing elements are used. The first row of the array consists of the basic elements of Fig. 2a (PE1). The left most processor in each row (PE1') is shown in Fig. 2b The processing elements shown in Fig. 2 (a) computes equation (12) while the elements in Fig. 2 (b) computes equation

the partial transform result. These processes are done in parallel, thereby consuming only one addition and one multiplication time equivalently. The other processing elements (PE2) are shown in Fig. 3 . They use the cosine values already computed in the previous processors PE1 or PE1'. PE2 is more simple (one multiplier + one adder) than PE1 but consumes the same computation time. This reduces the hardware complexity of the overall array. The upper left processing element (PE) is the only PE that computes equation (11) & (12) simultaneously. Although the number of multipliers and adders differ in each type of processing element, they still consume the same computation time (one addition and one multiplication).

To obtain the final value $\tilde{X}(k_1,k_2)$ an additional row of PE2 s with zero data sequence is introduced. The reason for choosing this type processors is to maintain the regularity of the array. $\tilde{X}(1,1),\tilde{X}(2,1)$, ... are sequentially produced from the right most processing element.

IV- Computation Complexity.

$\tilde{X}(k_1,k_2)$can be obtained after (N_1+N_2) computation steps. if we let M and A be the time for performing one multiplication and one addition, respectively, then M+A elapses in one processing element, i.e one computation step. Thus, $\tilde{X}(k_1,k_2)$ is available after $(N_1+N_2)(M+A)$ time units. For the sake of pipelining, the computation of the whole transformed sequence needs $\{(N_1+N_2)+(k_1+k_2)\}(M+A)$ units of time. The number of additions A(N,N) and multiplications M(N,N) needed to compute the (N x N) - point DCT using the proposed systolic array are

$$A(N \times N) = 4N$$
$$M(N \times N) = 4N$$

Thus the number of multiplications & additions for the proposed algorithm is O(N), while for the fast cosine transform algorithm {equation (27) in [1] } it is $O(N^2)$.

V-Conclusion:

For real - time processing, the hardware cost paid for the recursive computation is worth compared to the simple prestorage schemes for transform kernel values. The computations of the kernel values and the intermediate results are executed in parallel in each basic element, thus consuming only one multiplication and one addition at a time. This algorithm results in a significant reduction in computation time relative to the fast cosine transform implementation [1]

References

[1] Sc. Chan and K.L.HO, "A New Two - Dimensional Fast Cosine Transform Algorithm", IEEE Signal Processing, vol. 39, No. 2, pp. 481-485 Feb. 1991.

[2] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete cosine Cosine and Fourier Transforms ", IEEE Trans Comput., vol. C-23 , pp 90-94, Jan. 1974.

[3] M Hamidi and J. Pearl, "Comparison of the cosine and Fourier Transforms of Markov - I Signals, " IEEE Trans. Acoust., Speech, Signal Processing vol. ASSSP-24, pp. 428-429, Oct. 1976.

[4] H.T. Kung, "Why systolic architectures" Computer, pp. 37-46, Jan. 1992.

[5] S. Y. Kung, VLSI Array Processors, Prentice Hall, 1988.

[6] S. Manohar and G. Baudet "Pragmatic Approach to Systolic Design", IEE Proceeding, vol. 137, PL. E.No. 4, pp. 277 - 282 July 1990.

[7] H. S. Hou, "A Fast Recursive Algorithm For Computing the Discrete Cosine Transform, "IEEE Trans Acoust., Speech , Signal Processing, vol. ASSP - 35, pp. 1445 - 1461, Oct. 1987.

[8] 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.

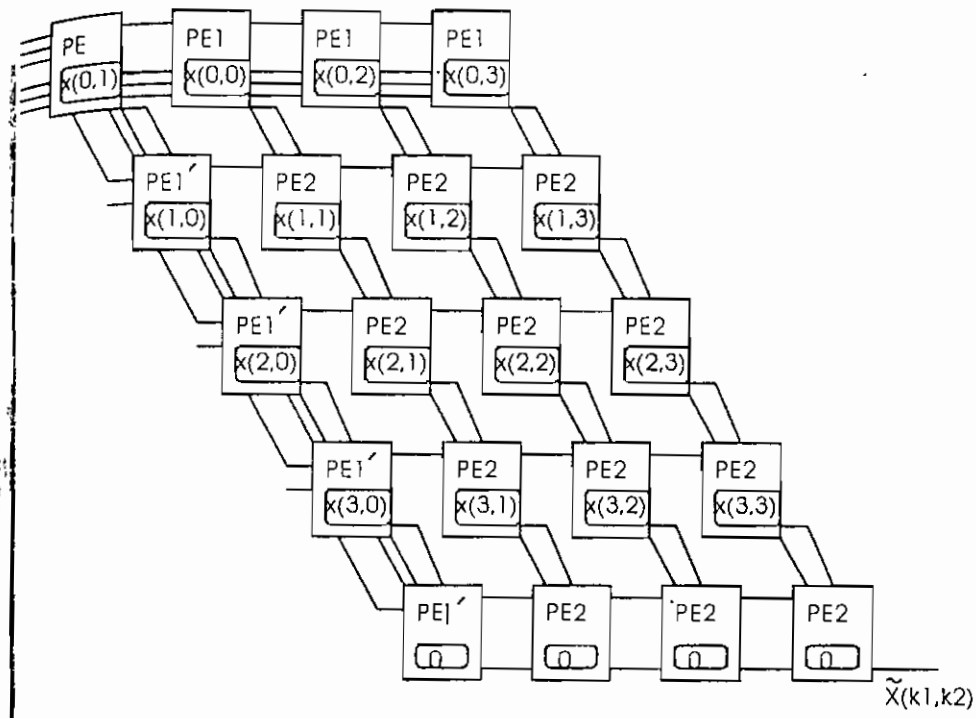
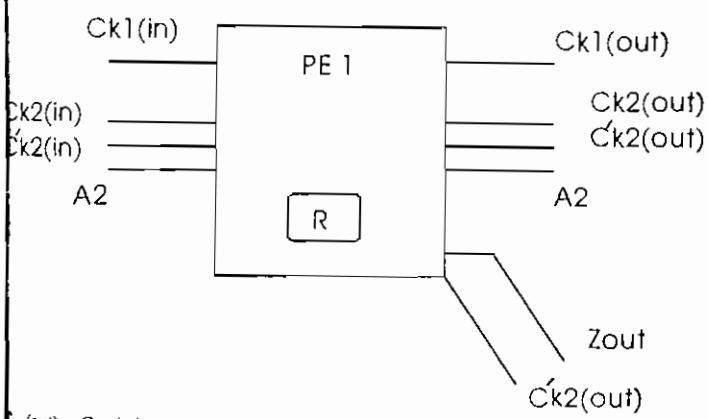
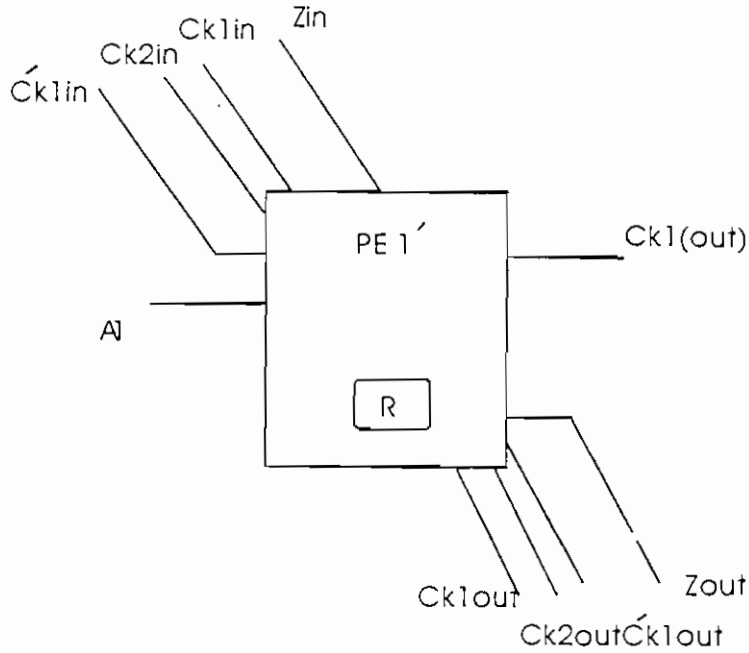


Fig. 1 The proposed systolic array for 2-D DCT (N_1=N_2=4)



$C_{k1}(out) = C_{k1}(in)$
$C_{k2}(out) = A_2 * C_{k2}(in) - C'_{k2}(in)$
$C'_{k2}(out) = C_{k2}(in)$
$Zout = R * C_{k1}(in)\ C_{k2}(in)$

Fig. 2(a) PE 1 (in the 1st row of the array) computes (12)



$C_{k1}out = C_{k1}in$
$C_{k2}out = C_{k2}in$
$C'_{k1}out = A_1C_{k1}in - C'_{k1}in$
$Zout = Zin + R^* C_{k1}inC_{k2}in$

Fig. 2 (b) PE 1´(the left most processing elements) computes(11)