

A NEW SYSTOLIC ALGORITHM OF 2-D DCT TRANSFORM BASED ON PSEUDO-CORRELATION STRUCTURES FOR A UNIFIED VLSI ARCHITECTURE

BY

DORU-FLORIN CHIPER*

Technical University “Gheorghe Asachi” of Iași,
Department of Electronics, Telecommunications and Information Technology

Received: May 7, 2015

Accepted for publication: May 20, 2015

Abstract. Using a new input sequence restructuring a new VLSI algorithm for 2-D discrete cosine transform (DCT) for an efficient unified VLSI architecture, with appealing topological features and high performances has been obtained. The new algorithm has a modular and regular computational structure and can be computed in parallel using two pseudo-correlation structures, resulting in a high throughput VLSI implementation. Using the proposed algorithm we can obtain an efficient VLSI architecture for 2-D DCT having a high speed at a low hardware complexity. The architecture that can be obtained has good topological properties as a regular and modular structure and local connections and can be used to obtain a unified implementation for 2-D DCT and DST.

Key words: discrete cosine transform; systolic arrays; VLSI algorithm.

1. Introduction

The discrete cosine transform (DCT) (Ahmed *et al.*, 1974; Jain, 1976; Jain, 1989) is an important tool in some digital signal processing applications. It is already known that DCT is a good approximation of the statistically-optimal Karhunen-Loeve transform (Jain, 1976; Jain, 1989). It is used in many digital signal processing applications such as speech and image transform coding (Jain,

*Corresponding author: *e-mail*: chiper@etc.tuiasi.ro

1989; Zhang, 2007), DCT based subband decomposition in speech and image compression (Chen, 2007), or video transcoding (Fung & Siu, 2006). It is also used in other important applications such as: block filtering (Martucci & Mersereau, 1993), feature extraction (Jadhav & Holambe, 2008), digital signal interpolation (Wang *et al.*, 1993), image resizing (Park & Park, 2006), transform-adaptive filtering (Pei & Tseng, 1996; Mayyas, 2005) and filter banks (Bergen, 2008).

It is well known that, for highly correlated images, DCT yields better results and for low correlation ones DST yields lower bit rates. So a unified VLSI implementation is highly desired.

Usually the transform length for DCT used in transform coding is 8 or 16. But there are other important applications where we are using larger values of the transform length are used. Also, we can reduce blocks artifacts using a prime transform length of 11 or 17 and an overlapping technique. In some applications a prime factor length is a more suitable transform length than a power of two (Tatsaki *et al.*, 1995) are, as it can be used in applications where the transform length is a composite number, were the factors are mutually prime. In this paper we propose a new VLSI algorithm for 2-D DCT of length $N = 11$, which is a prime factor length.

It is well known that 2-D DCT is a computational intensive algorithm that is difficult to be used in real-time applications. In order to be used in such applications it is necessary to design efficient hardware accelerators using the VLSI technology that can speed up the execution of this transform. In order to do this is necessary to reformulate existing algorithms for 2-D DCT in an appropriate manner or to design entirely new ones.

In designing VLSI algorithms it is necessary to take into consideration that the efficiency of a hardware algorithm is due especially to the communication complexity and then to the computational one due to the fact that data transfer plays a key role in a VLSI implementation.

Cycle convolution and circular correlation algorithms have remarkable advantages over other ones due to their efficient input/output and data transfer operations. These computational structures can be efficiently implemented in VLSI, using distributed arithmetic (White, 1989) or systolic arrays (Kung, 1982). Thus, several solutions have been proposed for a VLSI implementation of DCT based on cyclic convolution or circular correlation (Meher, 2006; Chipier *et al.*, 2007; Cheng & Parhi, 2006; Chipier & Ungureanu, 2011; Xie *et al.*, 2013).

The above mentioned advantages of the circular correlation from a VLSI implementation point of view can be extended to other structures such as, for example, skew-circular and pseudo-circular correlations.

In this paper we propose a new systolic array algorithm for 2-D DCT using two pseudo-correlation structures. These structures have a regular and modular form and can be computed in parallel resulting he a high throughput VLSI implementation. We have used an appropriate restructuring method of the 2-D DCT into such regular structures. All the advantages of a circular

correlation based implementation, such as regularity, modularity, low I/O cost and a reduced data management scheme can be obtained with the proposed computational structures.

The rest of the paper is organized as follows: in Section 2 an efficient formulation of 2-D DCT is presented, using an example for a 2-D DCT of length $N = 11$ that can be used to obtain a unified VLSI implementation. In Section 3 we discuss some considerations about a VLSI implementation of the proposed algorithm using the systolic array architectural paradigm. Conclusions are presented in Section 4.

2. Systolic Algorithm for 2-D DCT

The 2-D DCT is defined as:

$$X(k, l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} x(i, j) \cos[(2i+1)k\alpha] \cos[(2j+1)l\alpha] \quad (1)$$

with:

$$\alpha = \frac{\pi}{2N}. \quad (2)$$

where: $x(i, j)$, $(i, j = 1, \dots, N-1)$ is a pixel of the image and $X(k, l)$, $(k, l = 1, \dots, N)$ is a transform coefficient.

In eq. (1) we have dropped the constant coefficient before this sums in order to simplify our presentation. In the final VLSI architecture we can introduce a multiplier to scale the output with this constant.

In the literature there are presented several 2-D VLSI architectures for DST. Most of them use the row-column decomposition method. Some of them are using a direct method to compute forward or inverse 2-D DCT or DST.

We can reformulate (1) as follows:

$$[X_N] = [C_N][x_N][C_N]^T, \quad (3)$$

where $[C_N]$ is defined as:

$$[C_N]_{i,j} = \cos[(2i+1)j \cdot \alpha]. \quad (4)$$

To compute (3) we have to compute first

$$[Y_N] = [x_N] \cdot [C_N]^T, \quad (5)$$

along the rows of the input $[x_N]$. We can transpose the relation (5) to obtain:

$$[Y_N]^T = [C_N] \cdot [x_N]^T. \quad (6)$$

In order to make more clear our ideas we will consider as an example a 2-D DCT transform of length $N = 11$.

We can write (6) as follows:

$$\begin{bmatrix} Y(0,0) & Y(0,1) & \cdots & Y(0,10) \\ Y(1,0) & Y(1,1) & \cdots & Y(1,10) \\ \vdots & \vdots & \vdots & \vdots \\ Y(10,0) & Y(10,1) & \cdots & Y(10,10) \end{bmatrix} = \begin{bmatrix} \cos(1\alpha) & \cos(3\alpha) & \cdots & \cos(21\alpha) \\ \cos(2\alpha) & \cos(6\alpha) & \cdots & \cos(42\alpha) \\ \vdots & \vdots & \vdots & \vdots \\ \cos(11\alpha) & \cos(33\alpha) & \cdots & \cos(231\alpha) \end{bmatrix} \times \begin{bmatrix} x(0,0) & x(0,1) & \cdots & x(0,10) \\ x(1,0) & x(1,1) & \cdots & x(1,10) \\ \vdots & \vdots & \vdots & \vdots \\ x(10,0) & x(10,1) & \cdots & x(10,10) \end{bmatrix}. \quad (7)$$

Then we can compute a row from equation (7) as follows:

$$\begin{bmatrix} Y(0) \\ Y(1) \\ \vdots \\ Y(10) \end{bmatrix} = \begin{bmatrix} \cos(1\alpha) & \cos(3\alpha) & \cdots & \cos(21\alpha) \\ \cos(2\alpha) & \cos(6\alpha) & \cdots & \cos(42\alpha) \\ \vdots & \vdots & \vdots & \vdots \\ \cos(11\alpha) & \cos(33\alpha) & \cdots & \cos(231\alpha) \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(10) \end{bmatrix}, \quad (8)$$

where: $x(i)$, ($i = 0, 1, \dots, N-1$) is a real input sequence.

We will reformulate relation (8) as a parallel computational structure, based on pseudo-correlation structures, using only one input restructuring sequence as opposed to (Chipier *et al.*, 2005) where we have used two such auxiliary input sequences.

To illustrate our approach, we will consider an example with the length $N = 11$ and the primitive root $g = 2$.

First, we will introduce the following auxiliary input sequence $\{x_a(i)\}$: $i = 0, 1, \dots, N-1$ that is defined as follows:

$$x_a(N-1) = x(N-1), \quad (9)$$

$$x_a(i) = x(i) + x_a(i+1), \quad (10)$$

for $i = N-2, \dots, 0$.

Using this restructuring input sequence we can reformulate (8) as follows:

$$\begin{bmatrix} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \\ Y(6) \\ Y(7) \\ Y(8) \\ Y(9) \\ Y(10) \end{bmatrix} = x_a(0) \begin{bmatrix} \cos(\alpha) \\ \cos(2\alpha) \\ \cos(3\alpha) \\ \cos(4\alpha) \\ \cos(5\alpha) \\ \cos(6\alpha) \\ \cos(7\alpha) \\ \cos(8\alpha) \\ \cos(9\alpha) \\ \cos(10\alpha) \end{bmatrix} - 2 \begin{bmatrix} T(1)\sin(\alpha) \\ T(2)\sin(2\alpha) \\ T(3)\sin(3\alpha) \\ T(4)\sin(4\alpha) \\ T(5)\sin(5\alpha) \\ T(6)\sin(6\alpha) \\ T(7)\sin(7\alpha) \\ T(8)\sin(8\alpha) \\ T(9)\sin(9\alpha) \\ T(10)\sin(10\alpha) \end{bmatrix}, \quad (11)$$

and:

$$Y(0) = x_a(0). \quad (12)$$

We have obtained a new auxiliary output sequence $\{T(k): k=1,2,\dots,N-1\}$ that can be computed in parallel, using two pseudo-correlation structures as :

$$\begin{bmatrix} T(2) \\ T(4) \\ T(8) \\ T(6) \\ T(10) \end{bmatrix} = \begin{bmatrix} s(4) & s(8) & -s(5) & s(10) & -s(9) \\ s(8) & -s(5) & s(10) & -s(9) & -s(4) \\ -s(5) & s(10) & -s(9) & -s(7) & -s(8) \\ -s(10) & s(9) & s(7) & s(3) & -s(5) \\ -s(9) & -s(7) & -s(3) & s(6) & -s(10) \end{bmatrix} \begin{bmatrix} x_c(2) - x_c(9) \\ x_c(4) - x_c(7) \\ x_c(8) - x_c(3) \\ x_c(5) - x_c(6) \\ x_c(10) - x_c(1) \end{bmatrix}, \quad (14)$$

$$\begin{bmatrix} T(9) \\ T(7) \\ T(3) \\ T(5) \\ T(1) \end{bmatrix} = \begin{bmatrix} -s(4) & -s(8) & s(5) & s(10) & s(9) \\ -s(8) & s(5) & -s(10) & -s(9) & s(4) \\ s(5) & -s(10) & s(9) & -s(7) & s(8) \\ s(10) & -s(9) & -s(7) & s(3) & s(5) \\ s(9) & s(7) & s(3) & s(6) & s(10) \end{bmatrix} \begin{bmatrix} x_c(2) + x_c(9) \\ x_c(4) + x_c(7) \\ x_c(8) + x_c(3) \\ x_c(5) + x_c(6) \\ x_c(10) + x_c(1) \end{bmatrix}$$

where $s(i) = \sin(2i\alpha)$.

In equations (13) and (14) we have used two index mappings $\nu(i)$ and $\mu(i)$ to realize a partition into two groups of the permutation of indexes $\{1,2,3,4,5,6,7,8,9,10\}$. They are defined as follows:

$$\{\nu(i): 1 \rightarrow 2, 2 \rightarrow 4, 3 \rightarrow 8, 4 \rightarrow 6, 5 \rightarrow 10\},$$

$$\{\mu(i): 1 \rightarrow 9, 2 \rightarrow 7, 3 \rightarrow 3, 4 \rightarrow 5, 5 \rightarrow 1\}.$$

The signs of terms in equation (13) and (14) are given by the functions $\varepsilon(k,i)$ and $\gamma(k,i)$ that are defined respectively as follows:

$$\varepsilon(k,i) \text{ is defined by the matrix } \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix} \text{ and}$$

$$\gamma(k,i) \text{ is defined by the matrix } \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Equation (3) can be computed using N N -point DCT along the rows of the input $[X_N]$, obtaining $[Y_N] = [X_N] \cdot [C_N]^T$. Then, we compute N N -point DCTs along the columns of the matrix obtained from the row transformed $[X_N] = [C_N] [Y_N]$.

It is important to note that this decomposition method reduces the computation complexity with a factor of 4.

3. A VLSI Implementation Discussion

Using the algorithm presented in Section 2, we can obtain an efficient unified VLSI systolic array, as will be described bellow. First using a dependence-graph based synthesis procedure (Kung, 1998), we can project eqs. (13) and (14) to obtain two linear systolic arrays. As the sine kernel of this equations is similar with that obtained for a DST algorithm it is possible to unify the resulted VLSI architectures. These arrays represent the main part of the architecture used for the VLSI implementation of the derived algorithm. We can obtain a further reduction of the hardware complexity, using an appropriate hardware sharing method as that proposed by Chipier, (1998).

The obtained processing elements consists of a multiplier, an adder and some multiplexers used to manage the differences in sign in eqs. (13) and (14), respectively.

It is important to mention that the so called I/O bottle-neck can seriously limit the usefulness of this pipeline VLSI architecture. In order to avoid this limitation in the proposed architecture each input data is used in as many processing elements as possible. So, as we input a reduced number of

input elements we can obtain a low I/O cost, an aspect that could be very useful in designing systolic arrays where the so called I/O bottle-neck can seriously limit the usefulness of this concept. In order to place all I/O channels at one of the two extreme ends of each systolic array we have used a specific control mechanism, called the tag-control mechanism (Jen & Hsu, 1998). Using specific control bits, called tags, we can control the content of the internal registers using only I/O channels placed at the two ends of the linear systolic array.

The pre-processing stage is used to obtain the auxiliary input sequence $\{x_a(i); i = 0, 1, \dots, (N-1)\}$ in an appropriate order in order to be fed into the two systolic arrays. It realizes the computation of the auxiliary input sequence according to eqs. (9) and (10) and also an appropriate permutation of it. As the input sequence $\{x(i); i = 0, 1, \dots, (N-1)\}$ is used in eq. (10) in a reversed order it is also necessary to reorder the input sequence. All these permutations can be obtained using two RAMs with N words each.

The post-processing stage is used to put the auxiliary output sequence $\{T(k); k = 1, 2, \dots, N-1\}$ in a natural order and to obtain the final output sequence using the computations described by eq. (11).

4. Conclusions

In this paper it is proposed a new VLSI algorithm for 2-D DCT that can be used to obtain an efficient unified VLSI architecture with appealing features for a VLSI implementation and high speed performance. The proposed algorithm uses some modular and regular computational structures called pseudo-correlations that can be computed in parallel, thus resulting a high throughput VLSI implementation. The proposed systolic algorithm can be used to obtain an efficient VLSI architecture for 2-D DCT having a high speed at a low hardware complexity. The architecture that can be obtained has high performances and good topological properties and can be used to obtain a unified implementation for 2-D DCT and DST.

REFERENCES

- Ahmed N., Natarajan T., Rao K.R., *Discrete Cosine Transform*. IEEE Trans. Comput., **C-23**, 1, 90-94 (1974).
- Bergen S.W., *A Design Method for Cosine-Modulated Filter Banks Using Weighted Constrained-Least-Squares Filters*. Digital Signal Proc., **18**, 3, 282-290 (2008).
- Chen Y-Y., *Medical Image Compression Using DCT-Based Subband Decomposition and Modified SPIHT Data Organization*. Internat. J. of Medical Informatics, **76**, 717-725 (2007).
- Cheng C., Parhi K.K., *Hardware Efficient Fast DCT Based on Novel Cyclic Convolution Structures*. IEEE Trans. Signal Proc., **54**, 11, 4419-4434 (2006).

- Chipier D.F., *A New Systolic Array Algorithm for Memory-Based VLSI Array Implementation of DCT*. Proc. of IEEE Internat. Symp. on Computers and Commun., ISCC'97, July 1-3, 1997, 297-301.
- Chipier D.F., Swamy M.N.S., Ahmad M.O., *An Efficient Unified Framework for the VLSI Implementation of a Prime-Length DCT/IDCT with High Throughput*. IEEE Trans. on Signal Proc., Regular Papers, **54**, 6 (2007).
- Chipier D.F., Swamy M.N.S., Ahmad M.O., Stouraitis T., *Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST*. IEEE Trans. On Circ. A. Syst._I: Regular papers, **52**, 6 (2005).
- Chipier D.F., Ungureanu P., *Novel VLSI Algorithm and Architecture with Good Quantization Properties for a High-Throughput Area Efficient Systolic Array Implementation of DCT*. EURASIP J. on Advances in Signal Proc., **2011** (2011).
- Fung K-T., Siu W-C., *On Re-Composition of Motion Compensated Macroblocks for DCT-Based Video Transcoding*. Signal Proc.: Image Communication, **21**, 44-58 (2006).
- Jadhav D.V., Holambe R.S., *Random and Discrete Cosine Transform Based Features Extraction and Dimensionality Reduction Approach for Face Recognition*. Signal Proc., **88**, 2604-2609 (2008).
- Jain A.K., *A Fast Karhunen-Loeve Transform for a Class of Random Processes*. IEEE Trans. Comm., **COM-24**, 10, 1023-1029 (1976).
- Jain A.K., *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ, Prentice-Hall, 1989.
- Jen C.W., Hsu H.Y., *The Design of Systolic Arrays with Tag Input*. Proc. IEEE Int. Symp. Circuits and Systems ISCAS'88, Finland, 1988, 2263-2266.
- Kung H.T., *Why Systolic Architectures*. IEEE Comp., **15**, 37-46 (1982).
- Kung S.Y., *VLSI Array Processors*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.
- Martucci S.A., Mersereau R., *New Approaches to Block Filtering of Images Using Symmetric Convolution and the DST or DCT*. Proc. IEEE Internat. Symp. Circ. Syst. (ISCAS'93), May 1993, 259-262.
- Mayyas K., *A Note on Performance Analysis of the DCT-LMS Adaptive Filtering Algorithm*. Signal Proc., **85**, 1465-1467 (2005).
- Meher P.K., *Systolic Designs for DCT Using a Low-Complexity Concurrent Convolutional Formulation*. IEEE Trans. Circ. Syst. Video Technol. **16**, 9, 1041-1050 (2006).
- Park Y.S., Park H.W., *Arbitrary-Ratio Image Resizing Using Fast DCT of Composite Length for DCT-Based Transcoder*. IEEE Trans. Image Proc., **15**, 2, 494-500 (2006).
- Pei S.-C., Tseng C.-C., *Transform-Domain Adaptive Filter*. IEEE Trans. Signal Proc., **44**, 12, 3142-3146 (1996).
- Tatsaki A., Dre C., Stouraitis T., Goutis C., *Prime-Factor DCT Algorithms*. IEEE Trans. on Signal Proc., **43**, 3, 772-776 (1995).
- Wang Z., Jullien G.A., Miller W.C., *Interpolation Using the Discrete Sine Transform with Increased Accuracy*. Electron. Letters, **29**, 22, 1918-1920 (1993).

- White S.A., *Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review*. IEEE ASSP Mag., **6**, 3, 5-19 (1989).
- Xie J., Meher P.K., He J., *Hardware Efficient Realization of Prime-Length DCT Based on Distributed Arithmetic*. IEEE Trans. On Computers, **62**, 6, 1170-1178 (2013).
- Zhang D., Lin S., Zhang Y., Yu L., *Complexity Controllable DCT for Real-Time H.264 Encoder*. J. of Visual Commun. a. Image Representation, **18**, 59-67 (2007).

UN NOU ALGORITM SISTOLIC PENTRU TRANSFORMATĂ 2-D DCT BAZAT
PE STRUCTURI COMPUTAȚIONALE DE PSEUDO-CORELAȚIE PENTRU O
ARHITECTURĂ VLSI UNIFICATĂ

(Rezumat)

Este prezentat un nou algoritm VLSI pentru transformata 2-D DCT care poate fi folosit pentru a obține o arhitectură VLSI unificată eficientă având anumite caracteristici atrăgătoare pentru o implementare VLSI și performanțe de viteză ridicate. Algoritmul propus utilizează niște structuri computaționale modulare și regulate cunoscute sub denumirea de pseudo-corelații care pot fi calculate în paralel conducând la un throughput ridicat. Algoritmul sistolic propus poate fi utilizat pentru a obține o arhitectură VLSI eficientă pentru transformata 2-D DCT având performanțe de viteză ridicate la o complexitate hardware redusă. Arhitectura ce poate fi obținută are performanțe ridicate și proprietăți topologice bune și poate fi utilizată pentru a obține o implementare unificată pentru transformatele 2-D DCT și DST.

