

BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Volumul 62 (66), Numărul 3, 2016
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

NOVEL VLSI ARCHITECTURE FOR 1-D DCT TRANSFORM BASED ON PSEUDO-CORRELATION STRUCTURES

BY

DORU-FLORIN CHIPER*

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

Received: August 15, 2016

Accepted for publication: September 21, 2016

Abstract. Using a new VLSI algorithm for 1-D DCT a new VLSI architecture has been obtained that can be unified with that for 1-D DST. The proposed architecture has some appealing features, such as a high throughput with a low hardware complexity, and some attractive topological features, like modularity, regularity and local interconnections. These features have been obtained due to the fact that the VLSI algorithm used has a modular and regular computational structure and can be computed in parallel using two pseudo-correlation structures, thus resulting a high throughput VLSI implementation. Due to the fact that the VLSI algorithm is highly modular, regular and uses only local connections it was possible to efficiently use the systolic array architectural paradigm.

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 important for some digital signal processing applications, especially in digital compression due to the fact that it is a good approximation of the statistically-optimal Karhunen-Loeve transform (Jain, 1976; Jain, 1989). Other important digital signal processing applications of DCT are in speech and

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

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

Due to the fact that in some applications DCT yields better results, while for others DST is more suitable, a unified VLSI implementation is highly desirable.

Usually the transform length for DCT used in transform coding is 8 or 16. However, in some applications a prime factor is a more suitable transform length than a power of two (Tatsaki *et al.*, 1995), as it can be used in applications where the transform length is a composite number where the factors are mutually prime. In this paper we will use a new VLSI algorithm for 1-D DCT of length $N = 13$ which is a prime factor length.

Because 1-D DCT is a computationally intensive algorithm, a software implementation is difficult to be used in real-time applications, where it is necessary to use a hardware implementation, such as a hardware accelerator using the VLSI technology, that can speed up the execution of this transform by several orders of magnitude. In order to obtain an efficient hardware implementation, it is necessary to reformulate existing 1-D DCT in an appropriate manner or, even better, to design entirely new algorithms.

In designing VLSI algorithms, data moving plays a critical role in determining the performances of a VLSI algorithm. Data moving is even more important than computational complexity in establishing the performances of a VLSI implementation of such an algorithm. Therefore, it is important to use special computational structures in the reformulation of the VLSI algorithm.

Cycle convolution and circular correlation algorithms are among these computational structures having efficient input/output and data transfer operations leading to efficient VLSI implementations using distributed arithmetic (White, 1989) or systolic arrays (Kung, 1982). We can mention here some efficient solutions for a VLSI implementation of DCT based on cyclic convolution or circular correlation (Meher, 2006; Chiper *et al.*, 2007; Cheng & Parhi, 2006; Chiper & Ungureanu, 2011; Xie *et al.*, 2013).

Some of the mentioned advantages of the circular correlation or cycle convolution from a VLSI implementation point of view can be obtained using some other structures as for example skew-circular and pseudo-circular correlations.

In this paper we propose a new VLSI architecture which is derived based on a new efficient VLSI algorithm. Using a restructuring input sequence a regular, modular and parallel VLSI algorithm has been obtained that uses two pseudo-correlation structures. These structures have a regular and modular form and can be computed in parallel, thus resulting a high throughput VLSI implementation. Using this algorithm a regular, modular VLSI architecture has been obtained that has a high throughput.

The rest of the paper is organized as follows: in Section 2 a new restructuring of 1-D DCT is proposed using an example for a 1-D DCT of length $N = 13$. In Section 3 a new VLSI implementation of the proposed algorithm is derived based on the systolic array architectural paradigm. Conclusions are presented in Section 4.

2. A VLSI Algorithm for 1-D DCT

For a real input sequence $x(i) : i = 0, 1, \dots, N-1$, 1-D DCT is defined as:

$$Y(k) = \sqrt{\frac{2}{N}} \cdot \sum_{i=0}^{N-1} x(i) \cos[(2i+1)k\alpha], \text{ for } k = 0, \dots, N-1, \quad (1)$$

with

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

We will reformulate relation (1) as a parallel computational structure based on pseudo-correlation structures using only one input restructuring sequence as opposed to (Chiper *et al.*, 2005) where we have used two such auxiliary input sequences.

To illustrate our approach, we will consider an example of 1-D DCT with the length $N = 13$ 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:

$$x_a(N-1) = x(N-1) \quad (3); \quad x_a(i) = x(i) + x_a(i+1) \quad (4)$$

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

Using this restructuring input sequence we can reformulate (1) as follows:

$$\begin{bmatrix} Y(1) \\ Y(2) \\ Y(3) \\ Y(4) \\ Y(5) \\ Y(6) \\ Y(7) \\ Y(8) \\ Y(9) \\ Y(10) \\ Y(11) \\ Y(12) \end{bmatrix} = x_a(0) \begin{bmatrix} \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \\ \cos(\alpha) \end{bmatrix} - 2 \begin{bmatrix} T(1)\sin(\alpha) \\ T(2)\sin(2\alpha) \\ T(3)\sin(3\alpha) \\ T(4)\sin(4\alpha) \\ T(5)\sin(5\alpha) \\ T(6)\sin(6\alpha) \\ T(7)\sin(7\alpha) \\ T(8)\sin(8\alpha) \\ T(9)\sin(9\alpha) \\ T(10)\sin(10\alpha) \\ T(11)\sin(11\alpha) \\ T(12)\sin(12\alpha) \end{bmatrix}, \quad (5)$$

and:

$$Y(0) = x_a(0). \quad (6)$$

We have introduced 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 :

$$\begin{bmatrix} T(2) \\ T(4) \\ T(8) \\ T(10) \\ T(6) \\ T(12) \end{bmatrix} = \begin{bmatrix} s(4) & s(5) & -s(3) & s(6) & s(1) & -s(2) \\ s(5) & -s(3) & s(6) & s(1) & -s(2) & -s(4) \\ -s(3) & s(6) & s(1) & -s(2) & -s(4) & -s(5) \\ -s(6) & -s(1) & s(2) & s(4) & s(5) & -s(3) \\ s(1) & -s(2) & -s(4) & -s(5) & s(3) & -s(6) \\ -s(2) & -s(4) & -s(5) & s(3) & -s(6) & -s(1) \end{bmatrix} \begin{bmatrix} x_a(2) - x_a(11) \\ x_a(4) - x_a(9) \\ x_a(8) - x_a(5) \\ x_a(3) - x_a(10) \\ x_a(6) - x_a(7) \\ x_a(12) - x_a(1) \end{bmatrix} \quad (7)$$

$$\begin{bmatrix} T(11) \\ T(9) \\ T(5) \\ T(3) \\ T(7) \\ T(1) \end{bmatrix} = \begin{bmatrix} -s(4) & -s(5) & s(3) & s(6) & -s(1) & s(2) \\ -s(5) & s(3) & -s(6) & s(1) & s(2) & s(4) \\ s(3) & -s(6) & -s(1) & -s(2) & s(4) & s(5) \\ s(6) & s(1) & -s(2) & s(4) & -s(5) & s(3) \\ -s(1) & s(2) & s(4) & -s(5) & -s(3) & s(6) \\ s(2) & s(4) & s(5) & s(3) & s(6) & s(1) \end{bmatrix} \begin{bmatrix} x_a(2) + x_a(11) \\ x_a(4) + x_a(9) \\ x_a(8) + x_a(5) \\ x_a(3) + x_a(10) \\ x_a(6) + x_a(7) \\ x_a(12) + x_a(1) \end{bmatrix}, \quad (8)$$

where

$$s(i) = \sin(2i\alpha). \quad (9)$$

In eqs. (7) and (8) 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:

$$\{\nu(i) : 1 \rightarrow 2, 2 \rightarrow 4, 3 \rightarrow 8, 4 \rightarrow 10, 5 \rightarrow 6, 6 \rightarrow 12\}$$

$$\{\mu(i) : 1 \rightarrow 11, 2 \rightarrow 9, 3 \rightarrow 5, 4 \rightarrow 3, 5 \rightarrow 7, 6 \rightarrow 1\}$$

The signs of terms in equation (7) and (8) are given by the functions $\varepsilon(k, i)$ and $\gamma(k, i)$ that are defined respectively as follows:

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

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

3. A VLSI Implementation Using Systolic Arrays

To obtain the VLSI architecture from Fig. 1 we have used a data dependence-graph based synthesis procedure (Kung, 1988). Thus, we can project eqs. (7) and (8) to obtain two linear systolic arrays. As the sine kernel of this equations is similar with that obtained for a DST algorithm, the resulted architecture can be used also to implement 1-D DST. An important reduction of the hardware complexity can be obtain if we use an appropriate hardware sharing method as that proposed by Chiper, (1997).

The obtained systolic array represents the main part of the architecture used for the VLSI implementation of the 1-D DCT.

Analyzing Fig.1 we can observe that the input data enter the array and flow through it. When the input data $x_a(i) + x_a(j)$ and $x_a(i) - x_a(j)$ are flowing through the linear array and they meet the control tag "1", they are stored in that processing element. It can be observed that it is very important for this type of architecture that the right data meet other data in the right processing element at the right time.

The functionality of a processing element from Fig. 1 is shown in Fig. 2. One processing element contains two multipliers, two adders and some registers to store the input data. It also contains some multiplexers that are used to select the correct sign and to store, when it is appropriate, the data into registers using the control tags. In order to control the sign we have allocated two control tags for each processing element, as can be seen in Fig. 1.

Another important feature of the proposed architecture consists in using the input data in as many processing elements as possible. This is really important in reducing the so called I/O bottleneck. The I/O bottle-neck is important in designing pipeline architecture due to the fact that it can be the most important limiting factor of the speed performances.

In order to place the input data at one of the two extreme ends of the array we have used a special mechanism based on using the control tags (Jen &

1988). Thus, using some control tags we can control the content of the registers from the two ends of the array.

Besides the main core presented in Fig. 1 we have a pre-processing stage and a post-processing one.

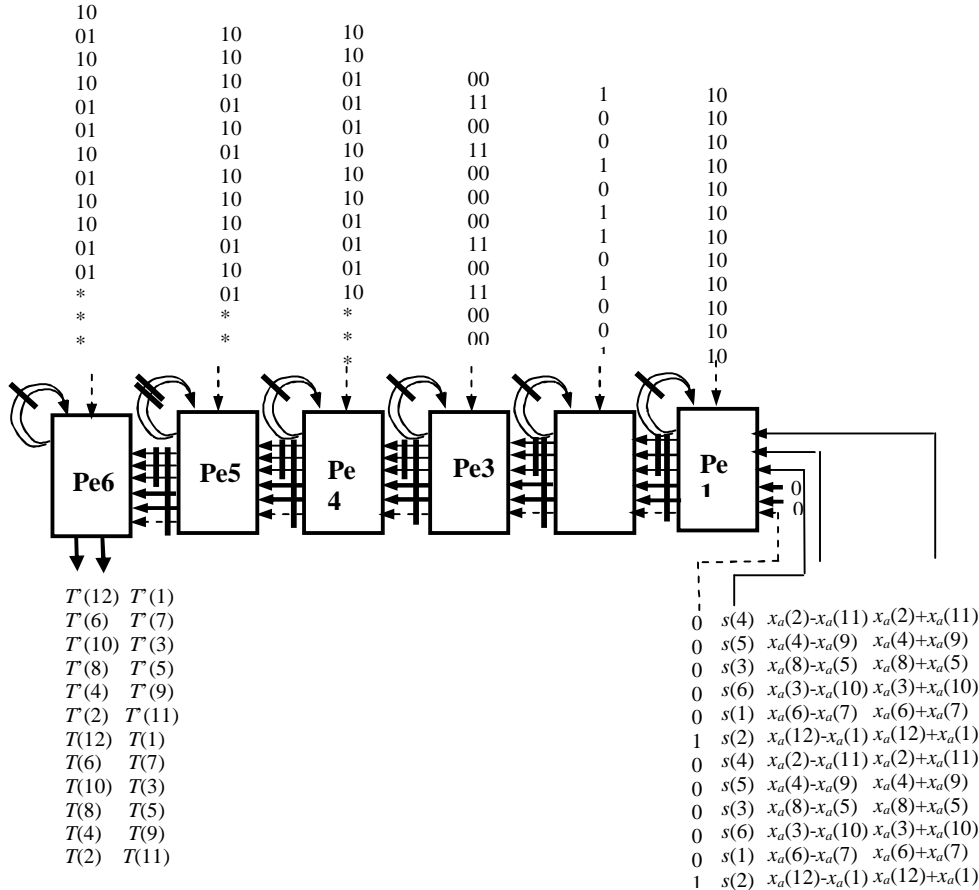


Fig. 1 – The VLSI architecture for 1-D DCT.

The pre-processing stage computes the auxiliary input sequence $\{x_a(i); i = 0, 1, \dots, (N - 1)\}$ using input data sequence $\{x(i); i = 0, 1, \dots, (N - 1)\}$ as can be seen in eqs. (2) and (4). Then the auxiliary input data is added and subtracted in order to obtain the necessary input sequences from Fig.1. Due to the fact that these input data are sent in a specific order it is necessary to permute them appropriately using two RAMs. The post-processing stage is used to permute the auxiliary output sequence $\{T(k) : k = 1, 2, \dots, N - 1\}$ and to obtain the final output sequence using equation (5). It can be seen that there are necessary two multipliers and two adders to compute the data from equation (5).

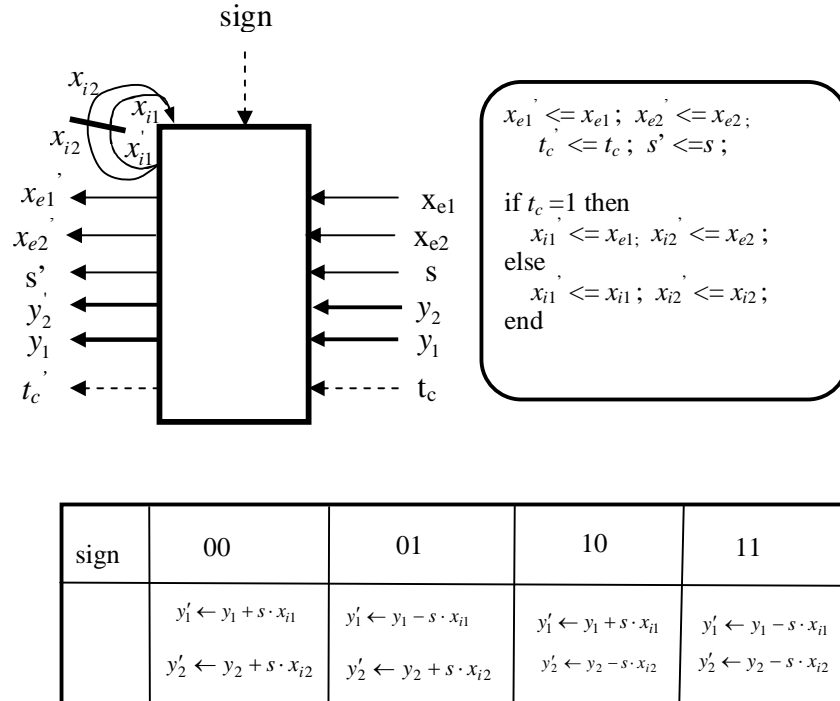


Fig. 2 – The functionality of a processing element.

4. Conclusions

In this paper a new VLSI architecture is proposed for 1-D DCT that can be efficiently unified with that for 1-D DST. The derivation of this architecture is based on a new VLSI algorithm for 1-D DCT that can be used to obtain an efficient implementation. The obtained VLSI architecture, besides a high throughput, has some appealing features like modularity, regularity and local interconnections. Some of these features have been obtained due to the fact that 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. Because the VLSI algorithm is highly modular, regular and uses only local connections, it was possible to efficiently use the systolic array architectural paradigm. The VLSI architecture that can be obtained has high performances and good topological properties and can be used to obtain a unified implementation for 1-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 Processing, **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 Processing, **54**, 11, 4419-4434 (2006).
- Chiper D.F., *A New Systolic Array Algorithm for Memory-Based VLSI Array Implementation of DCT*, Proc. of IEEE Internat. Symp. on Computers and Communications, ISCC'97, July 1-3, 1997, 297-301.
- Chiper 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).
- Chiper 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 Circuits and Systems_I: Regula papers, **52**, 6 (2005).
- Chiper 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 Processing, **2011** (2011).
- Fung K-T., Siu W-C., *On Re-Composition of Motion Compensated Macroblocks for DCT-Based Video Transcoding*, Signal Processing: Image Communication, **21**, 44-58 (2006).
- Jadhav D.V., Holambe R.S., *Randon and Discrete Cosine Transform Based Features Extraction and Dimensionality Reduction Approach for Face Recognition*, Signal Processing, **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, pp. 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 Jersay, 1988.
- Martucci S.A., Mersereau R., *New Approaches to Block Filtering of Images Using Symmetric Convolution and the DST or DCT*, Proc. IEEE Int. Symp. Circuits Systems (ISCAS'93), May 1993, pp.259-262.
- Mayyas K., *A Note on Performance Analysis of the DCT-LMS Adaptive Filtering Algorithm*, Signal Processing, **85**, 1465-1467 (2005).
- Meher P.K., *Systolic Designs for DCT Using a Low-Complexity Concurrent Convolutional Formulation*, IEEE Trans. Circuits 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 Process., **15**, 2, 494-500 (2006).

- Pei S.-C., Tseng C.-C., *Transform-Domain Adaptive Filter*, IEEE Trans. Signal Processing, **44**, 12, 3142-3146 (1996).
- Tatsaki A., Dre C., Stouraitis T., Goutis C., *Prime-Factor DCT Algorithms*, IEEE Trans. on Signal Processing, **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*, Journal of Visual Communication and Image Representation, **18**, 59-67 (2007).

O NOUĂ ARHITECTURĂ VLSI PENTRU TRANSFORMATĂ 1-D DCT BAZATĂ PE UTILIZAREA STRUCTURILOR DE PSEUDO-CORELAȚIE

(Rezumat)

În această lucrare se prezintă o nouă arhitectură VLSI pentru transformata 1-D DCT care poate fi unificată în mod eficient cu cea pentru transformata 1-D DST. Dezvoltarea acestei arhitecturi se bazează pe utilizarea unui nou algoritm VLSI pentru transformata DCT 1-D, care poate fi utilizat pentru a obține o implementare eficientă. Arhitectura VLSI astfel obținută, pe lângă o productivitate ridicată, are câteva caracteristici atrăgătoare cum ar fi modularitate, regularitate și interconexiuni cu caracter local. Câteva dintre aceste caracteristici au fost obținute datorită faptului că algoritmul VLSI folosit utilizează anumite structuri computaționale, cum ar fi cele de pseudo-corelație, care pot fi calculate în paralel rezultând astfel o implementare VLSI cu o productivitate înaltă. Datorită faptului că algoritmul VLSI folosit are un caracter ridicat de modularitate și regularitate și folosește interconexiuni cu caracter local, a fost posibilă utilizarea eficientă a paradigmei arhitecturale sistolice. Arhitectura VLSI ce poate fi astfel obținută are performanțe ridicate și bune proprietăți topologice și poate fi utilizată pentru obținerea unei implementări unificate pentru transformatele 1-D DCT și DST.

