

BULETINUL INSTITUTULUI POLITEHNIC DIN IAȘI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iași
Tomul LX (LXIV), Fasc. 4, 2014
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

A NEW VLSI ARCHITECTURE FOR 2-D DST TRANSFORM OF PRIME LENGTH

BY

DORU-FLORIN CHIPER*

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

Received: October 25, 2014

Accepted for publication: November 14, 2014

Abstract. Using a recently proposed VLSI algorithm for 2-D discrete sine transform (DST) an efficient VLSI architecture is proposed. This VLSI architecture has a modular and regular hardware structure and can compute in parallel thus resulting in high speed performances. The proposed architecture has been obtained by mapping the VLSI algorithm into two linear systolic arrays and combining them into a single linear systolic array having a high computing speed and low I/O cost with a small number of I/O channels placed at the two ends of the linear array. The presented architecture has all the advantages of the systolic array paradigm, like regular and modular structure with an interconnection topology appropriated for the VLSI technology.

Key words: discrete sine transform; systolic array; VLSI algorithm; VLSI architecture.

1. Introduction

The discrete cosine transform (DCT) and discrete sine transform (DST) (Ahmed *et al.*, 1974; Jain, 1976; Jain, 1989) are important functions in many digital signal processing applications. They were designed as good approximations of the statistically-optimal Karhunen-Loeve transform (Jain, 1976; Jain, 1989). They can also be used in speech and image transform coding

* *e-mail:* chipper@etti.tuiasi.ro

(Jain, 1989; Zhang *et al.*, 2007), DCT based subband decomposition in speech and image compression (Chen, 2007), or video transcoding (Fung & Siu, 2006) and some other important applications 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).

For high correlation images DCT yields better results and for low correlation ones DST yields a better compression.

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. Thus, there are in the literature several prime factor algorithms (PFA) for 1-D DST (Chiper *et al.*, 2003). PFA can split a $N = N_1 \times N_2$ point DST, where N_1 and N_2 are mutually prime, into a two-dimensional $N_1 \times N_2$ DST. DST is then applied for each dimension and the results are combined by means of input and output permutations. Also, it is possible to combine prime-factor algorithms for an efficient computation or implementation of the 1-D DST transform for composite-lengths (Kar & Rao, 1994).

For computational intensive algorithms as 2-D DST it is necessary to design application specific hardware that can speed up the execution of these algorithms or to reformulate them in an appropriate manner. The data movement and transfer play a key role in obtaining an efficient VLSI implementation, as has been observed for FFT. In reformulation of the algorithms for an efficient VLSI implementation it is necessary to take into account these considerations. In the recent years there was highlighted the fact that some modular and regular computation structures, as cyclic convolution and circular correlation algorithms have remarkable advantages over other ones due to their efficient input/output operations and data transfer (Chiper *et al.*, 2007; Xie *et al.*, 2013; Chiper & Ungureanu, 2011; Chiper, 1997). These computational structures can be efficiently implemented in VLSI using distributed arithmetic (White, 1989) or systolic arrays (Kung, 1982).

These advantages of the cyclic convolution and circular correlation computational structures can be extended to other structures as for example skew-cycle and pseudo-cycle convolutions.

In this paper we propose a new systolic array VLSI architecture for the implementation of 2-D DST taking advantage of some regular and modular structures. A high processing speed can be obtained by an extensive use of the pipelining mechanism. Using an appropriately mapping technique two linear systolic arrays can be obtained. These structures can compute in parallel, thus resulting in a high throughput VLSI implementation. Using a hardware sharing technique these two linear arrays can be merged into a single linear systolic array with a reduced hardware complexity.

The used VLSI algorithm is also appropriated for a memory-based implementation as will be discussed in section 3. All the advantages of a cyclic convolution based implementation 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 a low complexity formulation is presented for the computation of the 2-D DST transform with an example for a 2-D DST of length $N = 7$. In section 3 we discuss the VLSI architecture that efficiently implements the proposed algorithm using the systolic array architectural paradigm. Conclusions are presented in section 4.

2. Systolic Algorithm for 2-D DST

The 2-D DST for a $N \times N$ pixel block can be defined as follows:

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

where:

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

and $x(i,j)$, ($i, j = 0, 1, \dots, N-1$) is the pixel of an image with $Y(k,l)$, ($k, l = 1, \dots, N$) the transform coefficient.

To simplify our presentation, we have dropped the constant coefficient from the equation (1) that represents the definition of 2-D DST.

Several 2-D VLSI architectures for DST exist in the literature. Most of them use the row-column decomposition method and some of them are using the direct method to compute forward or inverse 2-D DCT or DST.

We can de express (1) in a matrix form as:

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

where $[S_N]$ is defined as:

$$[S_N]_{i,j} = \begin{cases} 1 & \text{for } i=0; \\ \sin[(2i+1)j\alpha] & \text{otherwise.} \end{cases} \quad (4)$$

To compute (3) we are using two steps:

First, we compute:

$$[Y_N] = [X_N][S_N]^T \quad (5)$$

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

$$[Y_N]^T = [S_N][X_N]^T. \quad (6)$$

Then we can compute:

$$[x_N] = [S_N][Y_N]^T$$

To illustrate our approach we consider a 2-D DST transform of length $N = 7$.

We will reformulate relation (6) as two pseudo-cycle convolutions using a single new input restructuring sequence as opposed to (Chipier *et al.*, 2005), where we have used two such auxiliary input sequences, and also the properties of DST kernel and the properties of the Galois Field of indexes.

To illustrate our approach, we will consider an example with the length $N = 7$ and the primitive root $g = 3$.

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

$$x_a(N-1) = x(N-1) \quad (7)$$

$$x_a(i) = (-1)^i x(i) + x_a(i+1) \quad (8)$$

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

Using this restructuring input sequence, we can reformulate (6) as:

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

and:

$$Y(0) = x_a(0). \quad (10)$$

The new auxiliary output sequence $\{T(k) : k = 1, 2, \dots, N-1\}$ can be computed as two pseudo-cycle convolutions using the fact that the transform length N is a prime number:

$$\begin{bmatrix} T(4) \\ T(2) \\ T(6) \end{bmatrix} = \begin{bmatrix} x_a(3) + x_a(4) & -(x_a(2) + x_a(5)) & -(x_a(6) + x_a(1)) \\ -(x_a(6) + x_a(1)) & -(x_a(3) + x_a(4)) & x_a(2) + x_a(5) \\ -(x_a(2) + x_a(5)) & -(x_a(6) + x_a(1)) & -(x_a(3) + x_a(4)) \end{bmatrix} \begin{bmatrix} \sin(4\alpha) \\ \sin(2\alpha) \\ \sin(6\alpha) \end{bmatrix}, \quad (11)$$

$$\begin{bmatrix} T(3) \\ T(5) \\ T(1) \end{bmatrix} = \begin{bmatrix} x_a(3) - x_a(4) & (x_a(2) - x_a(5)) & (x_a(6) - x_a(1)) \\ (x_a(6) - x_a(1)) & -(x_a(3) - x_a(4)) & -(x_a(2) - x_a(5)) \\ (x_a(2) - x_a(5)) & (x_a(6) - x_a(1)) & -(x_a(3) - x_a(4)) \end{bmatrix} \begin{bmatrix} \sin(4\alpha) \\ \sin(2\alpha) \\ \sin(6\alpha) \end{bmatrix}. \quad (12)$$

We have used two index mappings noted with $\xi(i)$ and $\zeta(i)$ that split into two groups the indexes $\{1,2,3,4,5,6\}$. They are defined as:

$$\{\xi(i) : 1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 6\}$$

$$\{\zeta(i) : 1 \rightarrow 3, 2 \rightarrow 5, 3 \rightarrow 1\}$$

The signs of the terms in equation (11) and (12) are given by the functions $\delta(k, i)$ and $\lambda(k, i)$ defined respectively as follows:

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

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

3. A New VLSI Architecture for DST

Eq. (3) can be implemented using the architecture from Fig. 1.

It computes a 2-D DST by $N \times N$ -point 1-D DST along the rows of the input $[X_N]$, obtaining $[Y_N] = [X_N][S_N]^T$, followed by $N \times N$ -point DSTs along the columns of the matrix obtained from the transformed row $[x_N] = [S_N][Y_N]$. The results of the first 1-D DST array have to be transposed in order to feed the next systolic array as shown in Fig.1. It can be observed that using the row-column decomposition method we have to compute two 1-D DSTs one after the other.

This simple decomposition method reduces the computation complexity by a factor of 4.

The 1-D DST systolic array from Fig. 1 can be implemented as follows: We can map the proposed algorithm into two linear systolic arrays that

implement eqs. (11) and (12) using a dependence-graph based synthesis procedure. These arrays can be merged into a single systolic array, as shown in Fig. 2, using a hardware sharing technique (Chiper *et al.*, 2005). The obtained

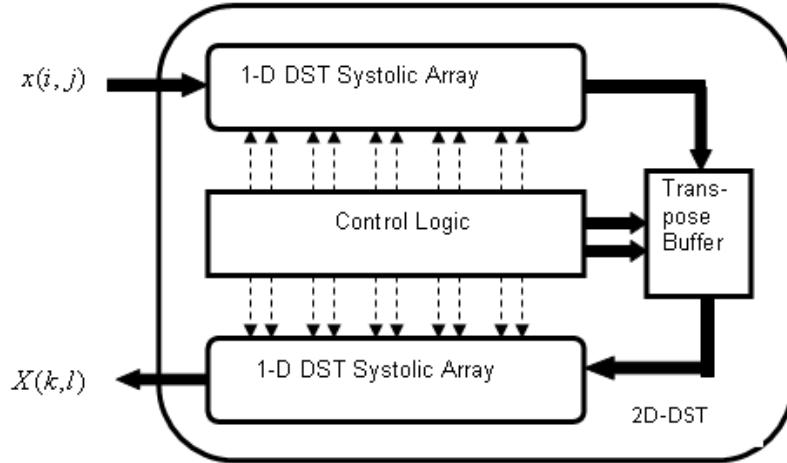
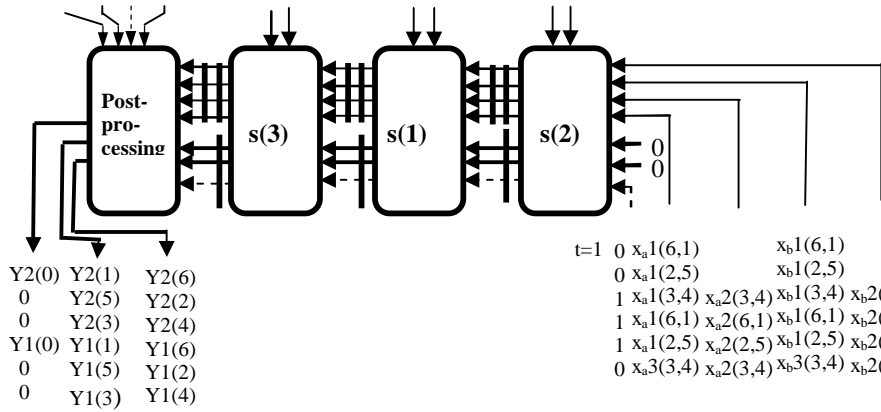


Fig. 1 – The linear systolic array for 2-D DST computation.

xa2(0) cs(a) 11 cs(6a)
 xa2(0) cs(5a) 10 cs(2a)
 xa2(0) cs(3a) 01 cs(4a)
 xa1(0) cs(a) 11 cs(6a)
 xa1(0) cs(5a) 10 cs(2a)
 xa1(0) cs(3a) 01 cs(4a)

1 1
 0 1 1 0
 1 0 1 1 1 0
 * * 1 0 1 0
 * * * * 0 0



$$x_a k(i, j) = x'_k(i) + x'_k(j) \text{ and } x_b k(i, j) = x'_k(i) - x'_k(j)$$

$$\text{with } a = \alpha \text{ and } cs(ka) : \{\cos(ka), \sin(ka)\}$$

Fig. 2 – Systolic array architecture for DST of length $N = 7$.

systolic array represents the hardware core of the architecture used for the VLSI implementation of the presented algorithm. The function of the processing elements PEs in the systolic array from Fig. 2 is presented in Fig. 3. The direct implementation of the processing elements defined in Fig. 3 results in two multipliers, some adders and also some multiplexers used to manage the differences in sign in eq. (11) and (12), respectively. Note that each multiplier achieves a multiplication with a constant in eqs. (11) and (12). This can be used to obtain an important improvement by replacing the multipliers with look-up tables LUTs residing in small ROMs.

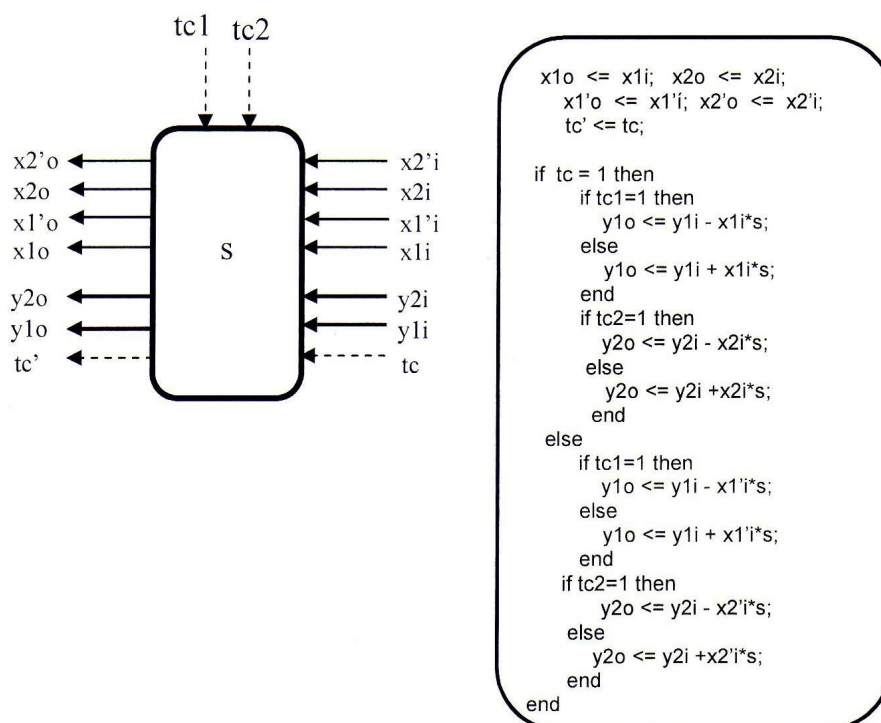


Fig. 3 – Functionality of a processing element PE in Fig. 2.

Due to the fact that eqs. (11) and (12) have the same form and the multiplications can be done with the same constant in each processing element, we can implement these multiplications with only one two-port ROM having a dimension of $2^{L/2}$ words. We can thus obtain a further reduction of the hardware complexity using an appropriate hardware sharing method. The clock cycle T is determined by $\max(T_{\text{Mem}}, T_A)$, where T_{Mem} represents the access time to the ROM memory and T_A is the latency of the adders from the critical path. The

actual value of the cycle period is determined by the value of the word length L and the implementation style for ROM and adders.

One important aspect of the presented solution is its low I/O cost. This feature is very useful in designing systolic arrays. It is important to note here that the so called I/O bottleneck can seriously limit the practical application of this architectural paradigm. The tag-control mechanism is used for systolic arrays to control the data-path presented in Fig. 2. This mechanism is appropriate for the systolic array architectural paradigm and is very useful in placing all I/O channels at the ends of each systolic array and to control the content of the internal registers using I/O channels placed at the two ends of the linear systolic array.

In the pre-processing stage we obtain the computation of the auxiliary input sequence using eqs. (7) and (8) and we realize the necessary permutations of the data sequences. The permutation blocks consist of a multiplexer and some latches and can permute the data sequences in a fully pipelined mode. Thus, the I/O data permutations are realized in such a manner that there is no time delay between the current and the next block of data. Each permutation can be also obtained using a RAM with N words.

The post-processing stage will reorder the auxiliary output sequence $\{T(k) : k = 1, 2, \dots, N - 1\}$ taking into consideration permutations $\xi(i)$ and $\zeta(i)$ in such a way as to put it in a natural order and to compute the final output sequence using eq. (9).

4. Conclusions

In this paper a new VLSI architecture for 2-D DST is presented, that leads to an efficient implementation using the actual VLSI technology. The proposed architecture has a modular and regular hardware structure with parallel processing thus resulting in a high performance VLSI implementation. It has been obtained by mapping the corresponding VLSI algorithm into two linear systolic arrays with a high throughput, low I/O cost and low hardware complexity and combining the two linear systolic arrays into a single linear systolic array using an appropriate hardware sharing technique. The obtained architecture is highly regular and modular and has local connections being well adapted to the VLSI technology.

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).
- Chiper D.F., *A New Systolic Array Algorithm for Memory-Based VLSI Array Implementation of DCT*. Proc. of IEEE Internat. Symp. on Computers a. Commun., 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 Circ. A. Syst.-I: Regula papers, **52**, 6 (2005).
- Chiper D.F., Swamy M.N.S., Ahmad M.O., Stouraits T., *A Systolic Array Architecture for the Discrete Sine Transform*. IEEE Trans. on Signal Proc., **50**, 9 (2002).
- 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 Adv. 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., *Randon 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.
- Kar D., Rao V.V.B., *On Prime Factor Decomposition Algorithm for the Discrete Sine Transform*. IEEE Trans. on Signal Proc., **42**, 11, 3258-3260 (1994).
- Kung H.T., *Why Systolic Architectures*. IEEE Comp., **15**, 37-46 (1982).
- Martucci S.A., Mersereau R., *New Approaches to Block Filtering of Images Using Symmetric Convolution and the DST or DCT*. Proc. IEEE Int. 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).
- 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 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 Acuracy*. Electron. Lett., **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).

O NOUĂ ARHITECTURĂ VLSI PENTRU TRANSFORMATA DST 2-D DE LUNGIME UN NUMĂR PRIM

(Rezumat)

Se prezintă o nouă arhitectură VLSI pentru transformata 2-D DST care conduce la o implementare eficientă folosind tehnologia VLSI actuală și având performanțe ridicate. Aceasta a fost obținută prin maparea algoritmului VLSI propus în două arii sistolice liniare având o viteză de procesare ridicată și un cost scăzut al operațiilor de intrare/ieșire și o complexitate hardware redusă și prin combinarea celor două arii sistolice într-o singură arie sistolică utilizând o tehnică de partajare hardware adecvată. Arhitectura astfel obținută are un grad ridicat de modularitate și regularitate fiind foarte adecvată tehnologiei VLSI actuale.