A New Linear Systolic Array for the VLSI Implementation of 2-D IDST

Doru Florin Chiper[1]

Abstract - In this paper a new linear VLSI array architecture for the VLSI implementation of the 2-D IDST based on a new systolic array algorithm is proposed. This new design approach uses a new efficient VLSI algorithm. It employs a new formulation of the inverse DST that is mapped on a linear systolic array. Using the proposed systolic array high computing speed is obtained with a low I/O cost. The proposed architecture is characterized by a small number of I/O channels located at the two extreme ends of the array together with a low I/O bandwidth that is independent of the transform length N. The topology of the proposed VLSI architecture is highly modular and regular and uses only local connections. Thus, it is well suited for a VLSI implementation

Keywords: Inverse discrete sine transform, systolic algorithms, systolic architectures

I. INTRODUCTION

The 2-D forward and inverse discrete sine transforms are important transform functions that are widely used in many signal and image processing applications. They are especially employed in image compression due to the fact that they behave very much like the statistically optimal Karhunen-Loeve transform (KLT). Thus, the forward and inverse 2-D DST and DCT represent the critical part in the implementation of JPEG compression [2].

The forward and inverse DST are computational intensive. So, in order to use them in real-time applications the development of application specific hardware is demanded.

In the literature there are presented several 2-D VLSI architectures [4-10]. Most of them use the row-column decomposition method. Some of them are using a direct method to compute forward or inverse 2-D DST or DCT [7-9].

Systolic arrays [11] are a good architectural paradigm to be used in real-time applications. They are also well suited for the VLSI implementation. The VLSI algorithms for forward and inverse DST have to be derived specifically. The way of data moving is very important in determination of the efficiency of a VLSI algorithm and of its implementation. Thus, the use of regular and modular computational structures

with local data communications can lead to efficient VLSI implementation [12, 13] using the systolic array architectural paradigm. Thus, an efficient way to convert the inverse DCT into such structures can lead to optimal VLSI implementations

II. TWO DIMENSIONAL IDST ARCHITECTURE

The 2-D inverse DST (IDST) for a NxN pixel block can be defined as follows:

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

where:

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

$x(k,l)$ $(k,l=0,1,...,N-1)$ is the pixel data, $X(i,j)$ $(i,j=0,1,...,N-1)$ is the transform coefficient .

In the literature there are presented several 2-D VLSI architectures for IDST. 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.

The row-column approach can de expressed in a matrix form as:

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

where $[S_N]$ is the 1-D N-point IDCT, with:

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

Equation (4) can be computed by N N-point IDST along the rows of the input $[X_N]$, obtaining $[Y_N]=[X_N][Y_N]^T$, and followed by N N-point IDSTs along the columns of the matrix obtained from the row transformed $[x_N]=[S_N][Y_N]$. It can be observed

[1] Facultatea de Electronică şi Telecomunicaţii, Bd. Carol I Nr. 11, 6600, Iasi.

that using the row-column decomposition method we have to compute two 1-D IDSTs one after the other.

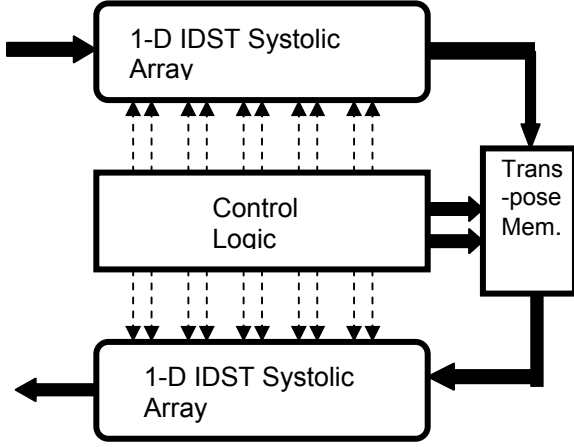This simple decomposition method reduces the computation complexity with a factor of 4.

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

III. 1-D N-POINT INVERSE DST ARCHITECTURE

A. Systolic Algorithm for 1-D Inverse DST

The 1-D N-point inverse discrete sine transform IDST is defined as follows:

$$x(k) = \sum_{i=0}^{N-1} Y(i) \cdot \sin[(2k+1)i \cdot \alpha]; \qquad (5)$$

$$k = 1,2,...,N$$

with $\alpha = \dfrac{\pi}{2N}$ (6)

In order to reformulate relation (5) as a circular correlation form we introduce some auxiliary sequences and use the proprieties of the Galois Field of indexes to appropriate permute the input and output sequences.

The output auxiliary sequence $\{T(k): k=1,2,...,N-1\}$ can be computed as follows:

$$T(k) = 2T'(k) \qquad (7)$$

The new auxiliary output sequence $\{T'(k): k=1,2,...,N-1\}$ can be computed as a circular correlation, if the transform length N is a prime number, as following:

$$T'(<g^k>_N) = \sum_{i=1}^{(N-1)/2} [(-1)^{\psi(k,i)} \cdot Y_C(<g^i>_N) +$$

$$+(-1)^{\psi(k,i+(N-1)/2)} \cdot Y_C(<g^{i+(N-1)/2}>_N)] \times \qquad (8)$$

$$\times \sin(<g^{i+k}>_N \times 2\alpha)$$

where $<x>_N$ denotes the result of x modulo N and

$$\psi(k,i) = \left\lfloor \frac{<g^k>_N \times <g^i> - <g^{i+k}>_N}{N} \right\rfloor \qquad (9)$$

with $\lfloor x \rfloor$ the greater integer smaller the x and is called the floor function.

We have used the properties of the Galois Field of indexes to convert the computation of the auxiliary output sequence $\{T'(k): k=1,2,...,N-1\}$ as a circular correlation.

The auxiliary input sequence $\{x_C(i): i=1,2,...,N-1\}$ is defined as following:

$$Y_C(i) = Y(i) \cdot \cos(i\alpha) \qquad (10)$$

Finally, the output sequence can be recursively computed using the auxiliary output sequence $\{T(k): k=1,2,...,N-1\}$ as:

$$x(k) = T(k) - x(k-1); \quad k=1,2,...,N-1 \quad (11)$$

$$x(0) = \sum_{i=1}^{N} Y_S(i) \qquad (12)$$

with

$$Y_S(i) = Y(i) \cdot \sin(i\alpha) \qquad (13)$$

B. An Example

To illustrate our approach, we will consider an example for 1-D IDST with the length N=11 and the primitive root g=2.

We can write (8) in matrix-vector product form as:

$$\begin{bmatrix} T'(2) \\ T'(4) \\ T'(8) \\ T'(5) \\ T'(10) \\ T'(9) \\ T'(7) \\ T'(3) \\ T'(6) \\ T'(1) \end{bmatrix} = \begin{bmatrix} s(4) & s(8) & s(5) & s(10) & s(9) \\ s(8) & s(5) & s(10) & s(9) & s(4) \\ s(5) & s(10) & s(9) & s(4) & s(8) \\ s(10) & s(9) & s(4) & s(8) & s(5) \\ s(9) & s(4) & s(8) & s(5) & s(10) \\ s(4) & s(8) & s(5) & s(10) & s(9) \\ s(8) & s(5) & s10) & s(9) & s(4) \\ s(5) & s(10) & s(9) & s(4) & s(8) \\ s(10) & s(9) & s(4) & s(8) & s(5) \\ s(9) & s(4) & s(8) & s(5) & s(10) \end{bmatrix} \times$$

$$\times \begin{bmatrix} \pm[x_C(2) \pm x_C(9)] \\ \pm[x_C(4) \pm x_C(7)] \\ \pm[x_C(8) \pm x_C(3)] \\ \pm[x_C(5) \pm x_C(6)] \\ \pm[x_C(10) \pm x_C(1)] \end{bmatrix}$$

$$(13)$$

where we noted by $s(k)$ as $\sin(2k\alpha)$ and the sign of the items in relation (9) is given by the following matrix:

$$SIGN = \begin{bmatrix} 01 & 01 & 11 & 01 & 11 \\ 01 & 11 & 01 & 11 & 11 \\ 11 & 01 & 11 & 11 & 11 \\ 00 & 10 & 10 & 00 & 00 \\ 11 & 11 & 11 & 01 & 11 \\ 10 & 10 & 00 & 00 & 00 \\ 10 & 00 & 10 & 10 & 00 \\ 00 & 10 & 00 & 10 & 00 \\ 11 & 01 & 01 & 01 & 11 \\ 00 & 00 & 00 & 00 & 00 \end{bmatrix}$$

where:

- The first bit designates the sign before the brackets
- The second bit denotes the sign inside the brackets

where the "1" bit indicates the minus sign (the first bit) and the subtraction operation (the second one)

III. THE LINEAR SYSTOLIC ARRAY FOR 1-D IDST

Using the dependence-graph of equation (13) and the dependence-graph based synthesis procedure [14] we have obtained a linear systolic array. The hardware-core of this array is presented in figure 2. The function of the processing elements Pes is presented in figure 2b. In order to deal with the sign differences in equation (13) we have used the tag-control technique presented in [15].

Using the tag-control mechanism we can keep the I/O channels at the two extreme ends of the linear array, where the tag sequences t_c controls the loading of the input data into the array as shown in fig.2b. Using this mechanism we can control the content of the internal registers using only channels placed at one of the two ends of the array.

The pre-processing and post-processing stages realize the appropriate reorder of the auxiliary input and output sequences.

In the preprocessing stage we also compute the auxiliary input sequence $\{Y_C(i) : i = 1,2,...,N-1\}$ and $\{Y_S(i) : i = 1,2,...,N-1\}$. In the post-processing stage we also compute the auxiliary output sequences $\{T(k) : k = 1,2,...,N-1\}$ and finally the output sequence using the equations (11), (12) respectively.

IV. PERFORMANCES AND COMPARISON

The average computation time is $(N-1)T_{cycle.}$ The number of multipliers is $(N-1)/2+1$ and the number of adders is $(N-1)/2+2$. Thus, low hardware and I/O costs can be obtained. We can easily obtain a high throughput using a two-level pipelining mechanism with low hardware and I/O costs.

In [16] a time-recursive structure is proposed. As compared with [16] the throughput is significantly increased using a two-level pipelining. The structure proposed in [16] did not allow a two-level pipelining due to its recursive nature.

In [17] and [18] the throughput can be also substantially increased using the two-level pipelining. These structures do not allow two level pipelining due to the data-path feedback.

As compared with [19] the throughput is also much increased when using a two-level pipelining. This is explained due to the presence of the feedback in RACs.

The proposed structured has also a low I/O cost. As compared with [20] the I/O cost is significantly lower. The I/O cost can significantly limit the speed performances due to so called I/O bottleneck.

V. CONCLUSION

In this paper a new VLSI architecture for the VLSI implementation of 2-D inverse discrete sine transform is presented. It has some appealing features as a low I/O cost and high speed performances. It employs a new VLSI algorithm that efficiently uses the advantages of the circular correlation computational structure as high degree of parallelism, small computational complexity and local data communications. The 2-D IDST VLSI architecture is obtained using two linear systolic arrays connected in a serial manner. The proposed VLSI architecture is highly regular and modular and has local interconnections. It has also a small number of I/O channels placed at the two extreme ends of the array with a reduced I/O bandwidth. Thus it is well suited for a VLSI implementation.

REFERENCES

[1] N. Ahmad, T.Natarajan, and K.R. Rao, "Discrete Cosine transform," IEEE Transactions on Computers, vol.C-23, pp.90-94, 1974.

[2] W. Pennebaker, J/ Michell. JPEG Still Image Data Compression Standard. Van Nostrand Reinhold, USA, 1992.

[3] M.Kovac, N. Ranganathan, "JAGUAR: A Fully Pipelined VLSI Architecture for JPEG Image Compression Standard," Proc. of IEEE, vol.83, No.2, 1995, pp.247-258.

[4] A. Madisseti, A. Wilson Jr., "A 100 Mhz 2-D 8x8 DCT/IDCT Processor for HDTV Applications," IEEE Transaction on Circuits and Systems for Video Technology, vol.5, no.2, pp. 158-165, Apr. 1995.

11

[5] S. Uramoto, et al. , "A 100 Mhz 2-D discrete cosine transform processor," IEEE Solid-State Circuits, 1992, vol.27, No.4, pp.492-498.

[6] M.T. Sun, T.C. Chen, and A.M. Gottlieb, "VLSI Implementation of 16x16 discrete cosine transform," IEEE Trans. on Circuits and Systems, 1989, vol.36, no.4, pp.610-617.

[7] C. Wang, C. Chen, "High-Throughput VLSI Architectures for the 1-D and 2-D Discrete Cosine Transform," IEEE Transaction on Circuits and Systems for Video Technology, vol.5, no.1, pp. 31-40, Febr. 1995

[8] Y. Lee, T. Chen, I. Chen, M. Chen, C.Ku, "A Cost-Effective Architecture for 8x8 Two-Dimensional DCT/IDCT Using Direct Method," IEEE Transaction on Circuits and Systems for Video Technology, vol.7, no.3, pp. 459-467, June 1997

[9] H.Lim, V.Piuri, E.E.Swartzlander, "A Serial-Parallel Architecture for Two-Dimensional Discrete Cosine and Inverse Cosine Transforms," IEEE Transactions on Computers, vol.49,No.12, pp.1297-1309, Dec. 2000.

[10] S. Bique, "New characterizations of 2D Discrete Cosine transform," IEEE Trans. on Computers, vol.54, no.9, Sept. 2005.

[11] H.T. Kung, "Why systolic architectures?," Computer Magazine, 1982, vol.15, no.1, pp.37-45.

[12] C.M. Rader, "Discrete Fourier transform when the number of data samples is prime," Proc. IEEE, vol.56, pp.1107-1108, June 1968.

[13] J.I. Guo, C.M. Liu and C.W. Jen, "A New Array Architecture for Prime-Length Discrete Cosine Transform," IEEE Transactions on Signal Processing, vol. SP-41,no.1, Jan. 1993.

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

[15] C.W. Jen and H.Y. Hsu, "The design of systolic arrays with tag input," Proc. IEEE Int. Symp. on Circuits and Systems, 1988.

[16] J. F. Yang and C-P. Fang, "Compact recursive structures for discrete cosine transform," IEEE Trans. on Circuits and Systems-II, vol. 47, pp. 314-321, Apr. 2000.

[17] W. H. Fang and M. L. Wu, "An efficient unified systolic architecture for the computation of discrete trigonometric transforms," in Proc. IEEE Symp. on Circuits and Systems, vol. 3, 1997, pp. 2092-2095.

[18] W. H. Fang and M-L. Wu, "Unified fully-pipelined implementations of one- and two-dimensional real discrete trigonometric transforms," IEICE Trans. on Fund. Electron. Commun. Comput. Sci., vol. E82-A, no. 10, pp. 2219-2230, Oct. 1999.

[19] J. Guo, C. Chen, and C-W. Jen, "Unified array architecture for DCT/DST and their inverses," Electron. Letters, vol. 31, no. 21, pp. 1811-1812, 1995.

[20] S.B.Pan and R-H. Park, "Unified systolic arrays for computation of DCT/DST/DHT," IEEE Trans. on Circuits and Systems for Video Technology, vol. 7, no. 2, pp. 413-419, April 1997.
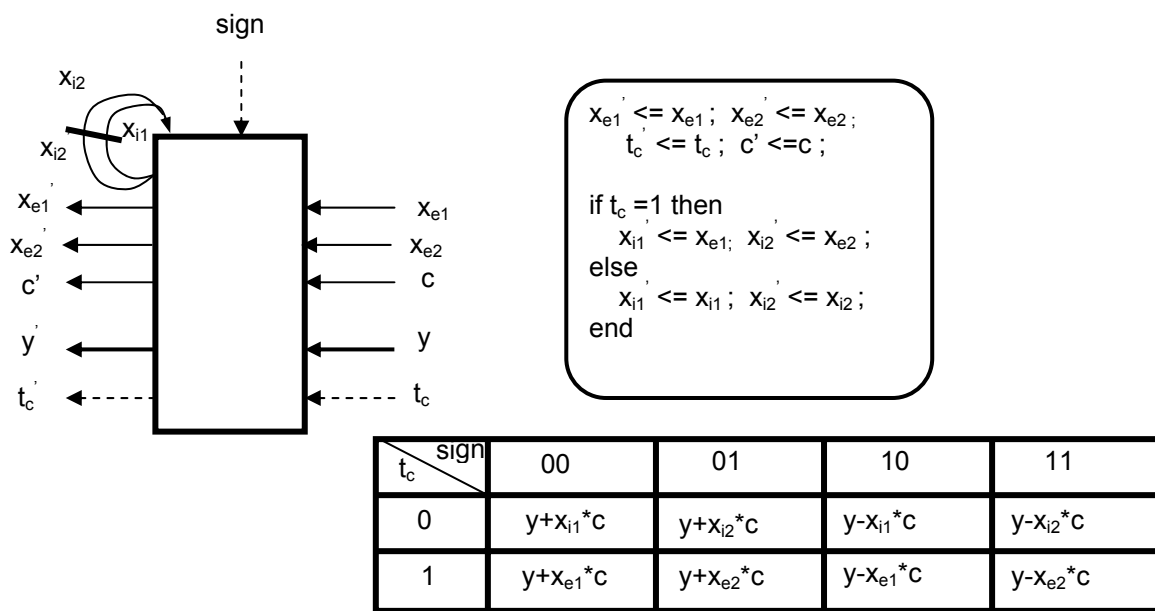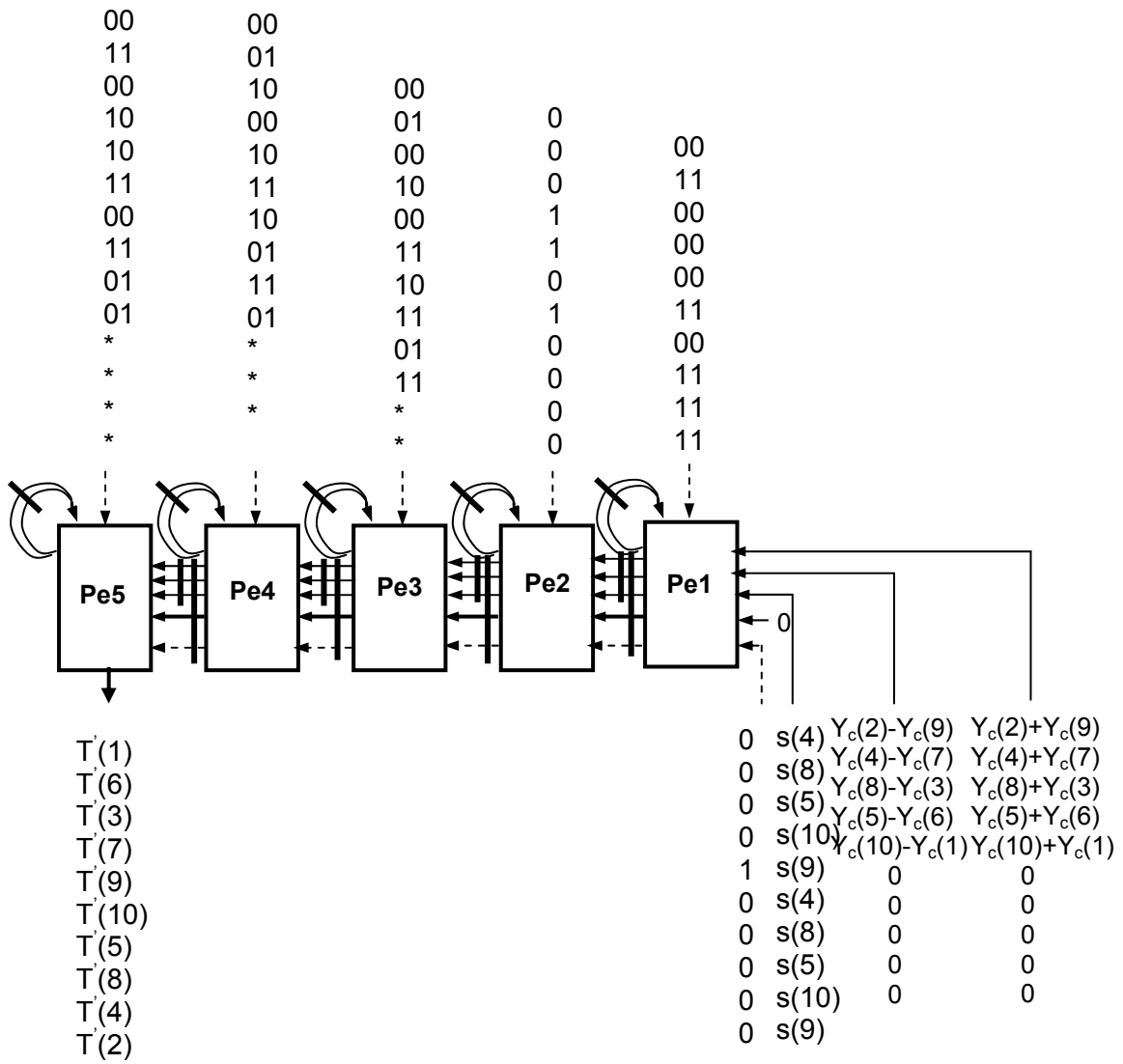
Fig.2. (a) The VLSI array architecture of the hardware-core of 1D-IDST

(b) The function of the processing elements PEs