Volume 6, Issue 2, February 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Architecture for Three Dimensional Type-II Discrete Sine Transform

M. N. Murty
Visiting Faculty, Department of Physics
Sri Sathya Sai Institute of Higher learning
Prasanthi Nilayam-515134
Andhra Pradesh, India

H. Panda
Department of Physics
S.K.C.G. (Autonomous) College
Paralakhemundi- 761200
Odisha, India

Abstract - In this paper, an algorithm for computation of three dimensional discrete sine transform (3D-DST) of even length N is proposed. Basing on this algorithm, a systolic architecture for implementation of 1D-DST for N = 4 is presented. In this algorithm, the number of multiplications of 1D-DST is reduced from N^2 to $N^2/2$. The computation of 3D-DST can be done by the row-column-frame decomposition technique. The proposed 3D-DST architecture consists of three identical 1D-DST modules and two sets of transpose registers.

Keywords- Discrete sine transform, Discrete cosine transform, Systolic architecture.

I. INTRODUCTION

Discrete transforms play a significant role in digital signal processing. Discrete cosine transform (DCT) and discrete sine transform (DST) are used as key functions in many signal and image processing applications. There are eight types of DCT and DST. Of these, the DCT-II, DST-II, DCT-IV, and DST-IV have gained popularity. The DCT and DST transform of types I, II, III and IV, form a group of so-called "even" sinusoidal transforms. Much less known is group of so-called "odd" sinusoidal transforms: DCT and DST of types V, VI, VII and VIII.

The DST was first introduced to the signal processing by Jain [1], and several versions of this original DST were later developed by Kekre*et al.*[2], Jain [3] and Wang *et al.* [4]. Ever since the introduction of the first version of the DST, the different DSTs have found wide applications in several areas in Digital signal processing (DSP), such as image processing[1,5,6], adaptive digital filtering[7] and interpolation[8]. The performance of DST can be compared to that of the DCT and it may therefore be considered as a viable alternative to the DCT. For images with high correlation, the DCT yields better results; however, for images with a low correlation of coefficients, the DST yields lower bit rates [9]. Yip and Rao [10] have proven that for large sequence length ($N \geq 32$) and low correlation coefficient ($\rho < 0.6$), the DST performs even better than the DCT.

The systolic architecture has the following characteristics:

- A massive and non-centralised parallelism
- Local communications
- Synchronous evaluation

The systolic arrays are used in the design and implementation of high performance digital signal processing equipment. Systolic architectures are established as the most popular and dominant class of VLSI structures due to the simplicity of their processing elements(PEs), modularity of their structure, regular and nearest neighbour interconnections between the PEs, high level of pipelinability, small chip area and lower dissipation. In the systolic architectures, the desired data are pumped rhythmically in regular intervals across the PEs for yielding high throughput by fully pipelined processing. The systolic array concept can also be exploited at bit level in the design of individual chips. The highly regular structure of systolic circuits renders them comparatively easy to design and test.

In this paper, an algorithm for computation of 3D-DST of even length *N*is presented**.** The proposed 3D-DST architecture consists of three identical 1D-DST modules and two sets of transpose registers.

The rest of the paper is organized as follows. The proposed algorithm for 3D-DST is presented in Section-II. The systolic architecture for computation of 1D –DST of length $N = 4$ is given in Section-III. Brief explanation of 3D-DST architecture is given in Section-IV. Conclusion is given in Section-V.

II. PROPOSED ALGORITHM FOR 3D-DST

The type-II 1D-DST for input data array $x(n), 1 \leq n \leq N$, is defined as

$$Y(k) = \sqrt{\frac{2}{N}} C_k \sum_{n=1}^{N} x(n) \sin\left[\frac{(2n-1)k\pi}{2N}\right] \quad (1)$$

$$\text{for } k = 1, 2, \ldots, N$$

where ,

$$C_k = \begin{cases} \frac{1}{\sqrt{2}} \; if \; k = N \\ 1 if \; k = 1, 2, ..., N-1 \end{cases}$$

The $Y(k)$ values represent the transformed data.

Hence, for a given 3D spatial input data sequence $\left\{ X_{ijk}; i, j, k = 1, 2, ..., N \right\}$, the 3D-DST output data sequence $\left\{ Y_{pqr}; p, q, r = 1, 2, ...N \right\}$ is written as

$$Y_{pqr} = C_p C_q C_r \sqrt{\frac{8}{N^3}} \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} X_{ijk} \sin\left[\frac{(2i-1)p\pi}{2N} \right] \sin\left[\frac{(2j-1)q\pi}{2N} \right] \sin\left[\frac{(2k-1)r\pi}{2N} \right] \quad (2)$$

Without loss of generality, the scale factor $C_p C_q C_r \sqrt{\frac{8}{N^3}}$ may be ignored in the rest of the paper. Also *N* is taken as even throughout the paper.

Denoting $\sin\left[\frac{(2h-1)l\pi}{2N} \right]$ by S_{lh} , (2) is represented as

$$Y_{pqr} = \sum_{k=1}^{N} \sum_{j=1}^{N} \sum_{i=1}^{N} X_{ijk} S_{pi} S_{qj} S_{rk}$$

$$= \sum_{k=1}^{N} \sum_{j=1}^{N} W_{pjk} S_{qj} S_{rk}$$

$$\Rightarrow Y_{pqr} = \sum_{k=1}^{N} T_{pqk} S_{rk} \quad (3)$$

where

$$W_{pjk} = \sum_{i=1}^{N} X_{ijk} S_{pi} \text{ for } p, j, k = 1, 2, ...N \quad (4)$$

and

$$T_{pqk} = \sum_{j=1}^{N} W_{pjk} S_{qj} \text{ for } p, q, k = 1, 2, ...N \quad (5)$$

In order to compute an $N \times N \times N$ -point DST (when *N* is even), three transforms (4), (5) and (3) need to be performed. When *N* is even, (4),(5) and (3) can be written as

$$W_{pjk} = \sum_{i=1}^{N/2} \left[X_{ijk} + (-1)^{p+1} X_{(N+1-i)jk} \right] S_{pi} \quad (6)$$

$$T_{pqk} = \sum_{j=1}^{N/2} \left[W_{pjk} + (-1)^{q+1} W_{p(N+1-j)k} \right] S_{qj} \quad (7)$$

$$Y_{pqr} = \sum_{k=1}^{N/2} \left[T_{pqk} + (-1)^{r+1} T_{pq(N+1-k)} \right] S_{rk} \quad (8)$$

Each of (6),(7) and (8) represents 1D-DST. By writing in this form, the number of multiplications in each1D-DST is reduced from N^2 to $N^2/2$.

The computation of 3D-DST Y_{pqr} is implemented by a cascade of three *N* -point 1D-DSTs as shown in the block diagram of Fig.1. First the 2D-DST is achieved by performing 1D-DST along the rows of input data array using (6) followed by the 1D-DST using (7) along the columns of the transformed array stored in transpose memory M_1. The outputs of 2D-DST are stored in a transpose memory M_2. Then, calculating the 1D-DST of the temporal frames using (8), the fully transformed output components Y_{pqr} are obtained.

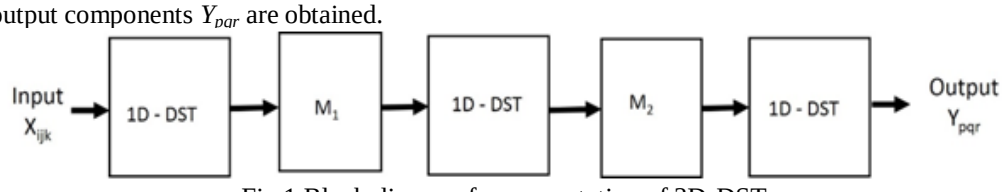


Fig.1 Block diagram for computation of 3D-DST

III. SYSTOLIC ARCHITECTURE FOR IMLEMENTATION OF 1D-DST W_{pjk} FOR $N = 4$

To clarify the proposed algorithm, 4-point 1D-DST is considered. Putting $N = 4$ and $p,j,k = 1,2,3,4$ in (6), we get the following expressions.

$$W_{111} = [X_{111} + X_{411}]S_{11} + [X_{211} + X_{311}]S_{12}$$
$$W_{211} = [X_{111} - X_{411}]S_{21} + [X_{211} - X_{311}]S_{22}$$
$$W_{311} = [X_{111} + X_{411}]S_{31} + [X_{211} + X_{311}]S_{32}$$
$$W_{411} = [X_{111} - X_{411}]S_{41} + [X_{211} - X_{311}]S_{42} \quad (9)$$
$$W_{121} = [X_{121} + X_{421}]S_{11} + [X_{221} + X_{321}]S_{12}$$
$$W_{221} = [X_{121} - X_{421}]S_{21} + [X_{221} - X_{321}]S_{22}$$
$$W_{321} = [X_{121} + X_{421}]S_{31} + [X_{221} + X_{321}]S_{32}$$
$$W_{421} = [X_{121} - X_{421}]S_{41} + [X_{221} - X_{321}]S_{42} \quad (10)$$
$$W_{131} = [X_{131} + X_{431}]S_{11} + [X_{231} + X_{331}]S_{12}$$
$$W_{231} = [X_{131} - X_{431}]S_{21} + [X_{231} - X_{331}]S_{22}$$
$$W_{331} = [X_{131} + X_{431}]S_{31} + [X_{231} + X_{331}]S_{32}$$
$$W_{431} = [X_{131} - X_{431}]S_{41} + [X_{231} - X_{331}]S_{42} \quad (11)$$
$$W_{141} = [X_{141} + X_{441}]S_{11} + [X_{241} + X_{341}]S_{12}$$
$$W_{241} = [X_{141} - X_{441}]S_{21} + [X_{241} - X_{341}]S_{22}$$
$$W_{341} = [X_{141} + X_{441}]S_{31} + [X_{241} + X_{341}]S_{32}$$
$$W_{441} = [X_{141} - X_{441}]S_{41} + [X_{241} - X_{341}]S_{42} \quad (12)$$
$$W_{112} = [X_{112} + X_{412}]S_{11} + [X_{212} + X_{312}]S_{12}$$
$$W_{212} = [X_{112} - X_{412}]S_{21} + [X_{212} - X_{312}]S_{22}$$
$$W_{312} = [X_{112} + X_{412}]S_{31} + [X_{212} + X_{312}]S_{32}$$
$$W_{412} = [X_{112} - X_{412}]S_{41} + [X_{212} - X_{312}]S_{42} \quad (13)$$
$$W_{122} = [X_{122} + X_{422}]S_{11} + [X_{222} + X_{322}]S_{12}$$
$$W_{222} = [X_{122} - X_{422}]S_{21} + [X_{222} - X_{322}]S_{22}$$
$$W_{322} = [X_{122} + X_{422}]S_{31} + [X_{222} + X_{322}]S_{32}$$
$$W_{422} = [X_{122} - X_{422}]S_{41} + [X_{222} - X_{322}]S_{42} \quad (14)$$
$$W_{132} = [X_{132} + X_{432}]S_{11} + [X_{232} + X_{332}]S_{12}$$
$$W_{232} = [X_{132} - X_{432}]S_{21} + [X_{232} - X_{332}]S_{22}$$
$$W_{332} = [X_{132} + X_{432}]S_{31} + [X_{232} + X_{332}]S_{32}$$
$$W_{432} = [X_{132} - X_{432}]S_{41} + [X_{232} - X_{332}]S_{42} \quad (15)$$
$$W_{142} = [X_{142} + X_{442}]S_{11} + [X_{242} + X_{342}]S_{12}$$
$$W_{242} = [X_{142} - X_{442}]S_{21} + [X_{242} - X_{342}]S_{22}$$
$$W_{342} = [X_{142} + X_{442}]S_{31} + [X_{242} + X_{342}]S_{32}$$
$$W_{442} = [X_{142} - X_{442}]S_{41} + [X_{242} - X_{342}]S_{42} \quad (16)$$

$$W_{113} = [X_{113} + X_{413}]S_{11} + [X_{213} + X_{313}]S_{12}$$
$$W_{213} = [X_{113} - X_{413}]S_{21} + [X_{213} - X_{313}]S_{22}$$
$$W_{313} = [X_{113} + X_{413}]S_{31} + [X_{213} + X_{313}]S_{32}$$
$$W_{413} = [X_{113} - X_{413}]S_{41} + [X_{213} - X_{313}]S_{42} \quad (17)$$
$$W_{123} = [X_{123} + X_{423}]S_{11} + [X_{223} + X_{323}]S_{12}$$
$$W_{223} = [X_{123} - X_{423}]S_{21} + [X_{223} - X_{323}]S_{22}$$
$$W_{323} = [X_{123} + X_{423}]S_{31} + [X_{223} + X_{323}]S_{32}$$
$$W_{423} = [X_{123} - X_{423}]S_{41} + [X_{223} - X_{323}]S_{42} \quad (18)$$
$$W_{133} = [X_{133} + X_{433}]S_{11} + [X_{233} + X_{333}]S_{12}$$
$$W_{233} = [X_{133} - X_{433}]S_{21} + [X_{233} - X_{333}]S_{22}$$
$$W_{333} = [X_{133} + X_{433}]S_{31} + [X_{233} + X_{333}]S_{32}$$
$$W_{433} = [X_{133} - X_{433}]S_{41} + [X_{233} - X_{333}]S_{42} \quad (19)$$
$$W_{143} = [X_{143} + X_{443}]S_{11} + [X_{243} + X_{343}]S_{12}$$
$$W_{243} = [X_{143} - X_{443}]S_{21} + [X_{243} - X_{343}]S_{22}$$
$$W_{343} = [X_{143} + X_{443}]S_{31} + [X_{243} + X_{343}]S_{32}$$
$$W_{443} = [X_{143} - X_{443}]S_{41} + [X_{243} - X_{343}]S_{42} \quad (20)$$
$$W_{114} = [X_{114} + X_{414}]S_{11} + [X_{214} + X_{314}]S_{12}$$
$$W_{214} = [X_{114} - X_{414}]S_{21} + [X_{214} - X_{314}]S_{22}$$
$$W_{314} = [X_{114} + X_{414}]S_{31} + [X_{214} + X_{314}]S_{32}$$
$$W_{414} = [X_{114} - X_{414}]S_{41} + [X_{214} - X_{314}]S_{42} \quad (21)$$
$$W_{124} = [X_{124} + X_{424}]S_{11} + [X_{224} + X_{324}]S_{12}$$
$$W_{224} = [X_{124} - X_{424}]S_{21} + [X_{224} - X_{324}]S_{22}$$
$$W_{324} = [X_{124} + X_{424}]S_{31} + [X_{224} + X_{324}]S_{32}$$
$$W_{424} = [X_{124} - X_{424}]S_{41} + [X_{224} - X_{324}]S_{42} \quad (22)$$
$$W_{134} = [X_{134} + X_{434}]S_{11} + [X_{234} + X_{334}]S_{12}$$
$$W_{234} = [X_{134} - X_{434}]S_{21} + [X_{234} - X_{334}]S_{22}$$
$$W_{334} = [X_{134} + X_{434}]S_{31} + [X_{234} + X_{334}]S_{32}$$
$$W_{434} = [X_{134} - X_{434}]S_{41} + [X_{234} - X_{334}]S_{42} \quad (23)$$
$$W_{144} = [X_{144} + X_{444}]S_{11} + [X_{244} + X_{344}]S_{12}$$
$$W_{244} = [X_{144} - X_{444}]S_{21} + [X_{244} - X_{344}]S_{22}$$
$$W_{344} = [X_{144} + X_{444}]S_{31} + [X_{244} + X_{344}]S_{32}$$
$$W_{444} = [X_{144} - X_{444}]S_{41} + [X_{244} - X_{344}]S_{42} \quad (24)$$

The systolic architecture for computing 1D-DST (6) for $N = 4$ is shown in Fig.2. It consists of $N/2$ adder/ subtractor cells and $N^2/2$ PEs for computing W_{pjk} given by (6). The function of each adder/subtractor cell is shown in Fig.3 and the function of each PE is shown in Fig.4. The pair of inputs X_{ijk} and $X_{(N+1-i)jk}$ enters the i^{th} adder/subtractor cell at the same

time. The pair of inputs X_{ijk} and $X_{(N+1-i)jk}$ enters the adder/subtractor cell one cycle ahead of the pair $X_{(i+1)jk}$ and $X_{(N-i)jk}$. The 0s in Fig.2 denote the delays. Each PE consists of one multiplier, one adder and one register for storing S_{pi}. As each PE contains one multiplier and there are $N^2/2$ PEs, the number of multiplications required for realizing 1D-DST is $N^2/2$ .When the input data move down the architecture, W_{pjk} given by the sets of expressions from (9) to (24) will be generated as shown in Fig.2.

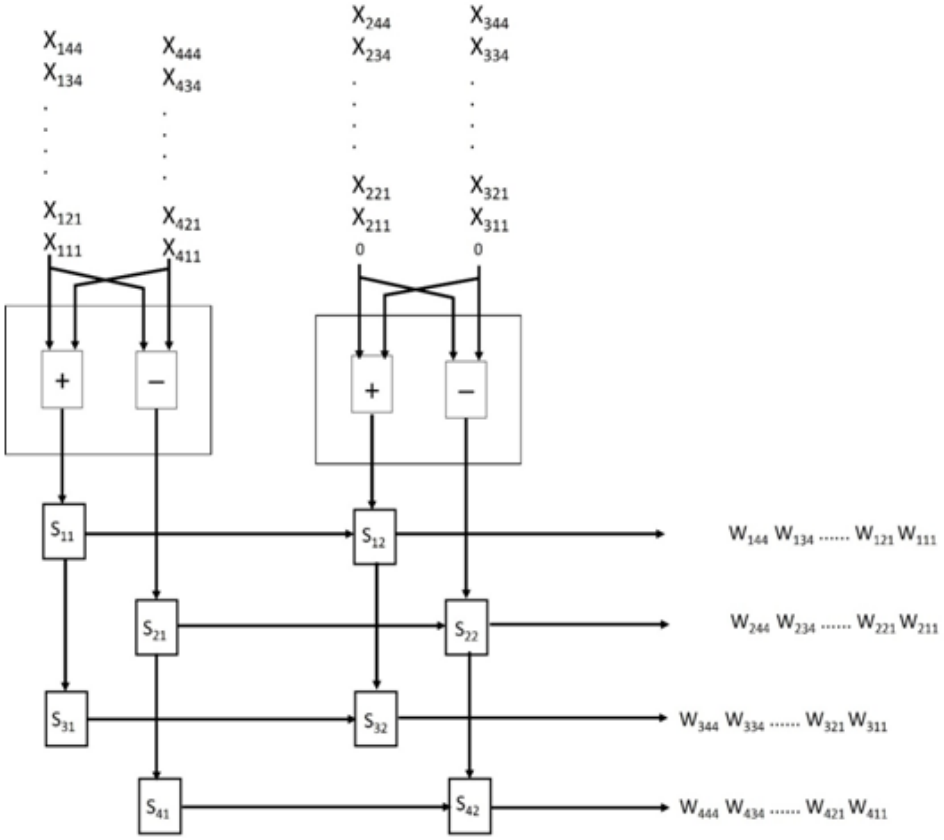


Fig.2 ID-DST Systolic architecture for $N = 4$

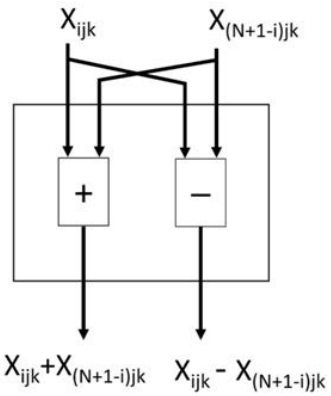
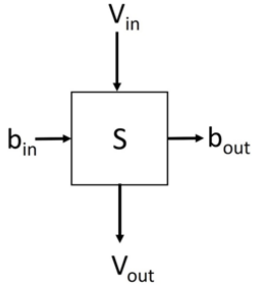


Fig.3 Adder/Subtractor cell of systolic architecture for ID-DST



$$b_{out} = b_{in} + SV_{in}$$

$$V_{out} = V_{in}$$

Fig.4 Function of each PE of systolic architecture for ID-DST

IV. 3D-DST ARCHITECTURE IN BRIEF

The parallel architecture for the computation of 3D-DST is based on the row-column-frame decomposition technique. This architecture uses the same 1D-DST module, shown in Fig.2, for each dimension and is divided into two main stages. Stage one and two computes the 2D-DST and 1D-DST respectively.

In stage one, the 2D-DST is computed according to the row-column separability technique. The 2D-DST architecture consists of two 1D-DST modules and a transposition matrix, which is implemented using two sets of skewed registers and the data is transferred from one set to the other through multiplexers.

In the second stage, the 2D-DST transformed coefficients are stored into the $1\times N$ register array. Then they are fed simultaneously into the final 1D-DST module, which evaluates the 3D-DST components.

V. CONCLUSION

In this paper, an algorithm for realizing 3D-DST of even length N is proposed. In this algorithm, the number of multiplications of 1D-DST is reduced from N^2 to $N^2/2$. Basing on this algorithm, a systolic architecture for implementation of 1D-DST for $N = 4$ is presented. This architecture utilizes parallel structures to achieve high speed performance. The proposed architecture for 3D-DST consists of three identical 1D-DST modules and two sets of transpose registers. The 3D-DST architecture is suitable for 2D or 3D video coding. The highly regular structure of systolic circuits renders them comparatively easy to design and test. The systolic architecture is suitable for VLSI implementation.

REFERENCES

[1] A.K. Jain, "A fast Karhunen-Loeve transform for a class of random processes", *IEEE Trans.Commun*., vol. COM-24,pp.1023-1029, September 1976.

[2] H.B. Kekre and J.K. Solanka, " Comparative performance of various trigonometric unitary transforms for transform image coding", *Int. J. Electron*., vol. 44,pp.305-315,1978.

[3] A.K. Jain, "A sinusoidal family of unitary transforms", *IEEE Trans. Patt. Anal. Machine Intell*., vol. PAMI-I,pp.356-365, September 1979.

[4] Z. Wang and B. Hunt, "The discrete W transform", *Applied Math Computat*., vol. 16, pp.19- 48,January 1985.

[5] S. Cheng, "Application of the sine transform method in time of flight positron emission image reconstruction algorithms", *IEEE Trans. BIOMED. Eng*., vol. BME-32,pp.185-192, March 1985.

[6] K. Rose, A. Heiman, and I. Dinstein, " DCT/DST alternate transform image coding", *Proc. GLOBECOM 87*, vol. I,pp.426-430, November 1987.

[7] J.L. Wang and Z.Q. Ding, "Discrete sine transform domain LMS adaptive filtering", *Proc. Int. Conf.Acoust., Speech, Signal Processing*,pp.260-263, 1985.

[8] Z. Wang and L. Wang, "Interpolation using the fast discrete sine transform", *Signal Processing*, vol. 26,pp.131-137, 1992.

[9] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs, NJ: Prentice - Hall, 1989.

[10] P. Yip and K.R. Rao, "On the computation and the effectiveness of discrete sine transform", *Comput. Electron.*,vol. 7,pp.45-55,1980.

