RECURSIVE ALGORITHMS FOR REALIZATION OF ONE-DIMENSIONAL DISCRETE SINE TRANSFORM AND INVERSE DISCRETE SINE TRANSFORM

M.N. Murty
Department of Physics, National Institute of Science & Technology,
Palur Hills, Berhampur-761008, Odisha, India.
Email: mnarayanamurty@rediffmail.com

ABSTRACT

In this paper, novel recursive algorithms for realization of one-dimensional discrete sine transform (DST) and inverse discrete sine transform (IDST) of any length are proposed. By using some mathematical techniques, recursive expressions for DST and IDST have been developed. Then, the DST and IDST are implemented by recursive filter structures. A linear systolic architecture for realization of DST is also presented in this paper. Compared with some other algorithms, the proposed algorithm for DST achieves savings on the number of multiplications and additions. The recursive algorithms have been found very effective for realization using software and VLSI techniques.

Keywords: *Discrete sine transform, Inverse discrete sine transform, Recursive algorithm, Systolic architecture.*

1. INTRODUCTION

The Discrete sine transform (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]. There exist four even DST's and four odd DST's, indicating whether they are an even or an odd transform[5]. Ever since the introduction of the first version of the DST, the different DST's have found wide applications in several areas in Digital signal processing (DSP), such as image processing[1,6,7], adaptive digital filtering[8] and interpolation[9]. The performance of DST can be compared to that of the discrete cosine transform (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 [10]. Yip and Rao [11] 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.

In this paper, two algorithms to convert 1-D DST and IDST of any size into recursive forms are presented. The DST and IDST are implemented by recursive filter structures. A systolic architecture for realization of 1-D DST of arbitrary length is presented. The proposed approach requires N multiplication and ($2N$-2) additions for realization of N length DST. The number of multiplications and additions in the proposed algorithm for DST are less in comparison with some existing structures [12] – [20].

The rest of the paper is organized as follows: The derivation of recursive algorithm for 1-D DST is presented in Section 2. An example for realization of 1-D DST is given in Section 3. The systolic architecture for computation of DST is presented in Section 4. The comparison of proposed realization of DST with other research works is presented in Section 5. The recursive algorithm for IDST is given in Section 6. An example for realization IDST is presented in Section 7. The conclusion is given in Section 8. Finally, references are given in Section 9.

2. PROPOSED RECURSIVE ALGORITHM FOR 1-D DST

Let $X(n),\ 1 \leq n \leq N$, be the input data array, then the type-II DST is defined as

$$Y(k) = \sqrt{\frac{2}{N}} C_k \sum_{n=1}^{N} X(n) \sin\left[\frac{k(2n-1)\pi}{2N}\right] \quad \text{for } k = 1, 2, \ldots, N \tag{1}$$

where $C_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = N \\ 1 & \text{Otherwise} \end{cases}$

The *Y* values represent the transformed date. Without loss of generality, the scale factors may be ignored in the rest of the paper.

Taking $z = \pi k / N$, (1) can be written as

$$Y(k) = \sum_{n=1}^{N} X(n) \sin\left[\left(n - \frac{1}{2}\right)Z\right] \qquad (2)$$

The recursive kernel a_r for realization of DST is introduced as given below:

$$a_r \sin Z = \sum_{n=r+1}^{N} X(n) \sin[(n-r)Z] \qquad (3)$$

$$= X(r+1) \sin Z + \sum_{n=r+2}^{N} X(n) \sin[(n-r)Z]$$

$$= X(r+1) \sin Z + \sum_{n=r+2}^{N} X(n) \{\sin[(n-r-1)Z]\cos Z + \cos[(n-r-1)Z]\sin Z\}$$

$$= X(r+1) \sin Z + \sum_{n=r+2}^{N} X(n) \{2\sin[(n-r-1)Z]\cos Z - \sin[(n-r-1)Z]\cos Z + \cos[(n-r-1)Z]\sin Z\}$$

$$= X(r+1) \sin Z + 2\sum_{n=r+2}^{N} X(n) \sin[(n-r-1)Z]\cos Z - \sum_{n=r+2}^{N} X(n)\sin[(n-r-2)Z] \qquad (4)$$

From (3), we have

$$a_{r+1} \sin Z = \sum_{n=r+2}^{N} X(n) \sin[(n-r-1)Z] \qquad (5)$$

and

$$a_{r+2} \sin Z = \sum_{n=r+3}^{N} X(n) \sin[(n-r-2)Z] \qquad (6)$$

Using (5) and (6), (4) can be written as

$$a_r \sin Z = X(r+1)\sin Z + 2a_{r+1} \sin Z \cos Z - a_{r+2} \sin Z \qquad (7)$$

Hence,

$$a_r = X(r+1) + 2a_{r+1} \cos Z - a_{r+2} \qquad (8)$$

for $r = 0, 1, 2, ..., N-1$ and $a_r = 0$ if $r \geq N$

Multiplying both sides of (2) by $\sin Z$, we get

$$Y(k)\sin Z = \sum_{n=1}^{N} X(n)\sin\left(nZ - \frac{Z}{2}\right)\sin Z$$

$$= X(1)\sin\left(\frac{Z}{2}\right)\sin Z + \sum_{n=2}^{N} X(n)\sin\left(nZ - \frac{Z}{2}\right)\sin Z$$

$$= X(1)\sin\left(\frac{Z}{2}\right)\sin Z + \sum_{n=2}^{N} X(n)\left\{\sin(nZ)\cos\left(\frac{Z}{2}\right) - \cos(nZ)\sin\left(\frac{Z}{2}\right)\right\}\sin Z$$

$$= X(1)\sin\left(\frac{Z}{2}\right)\sin Z + \sum_{n=2}^{N} X(n)\left\{\sin(nZ)\sin Z\ \cos\left(\frac{Z}{2}\right) - \sin(nZ)\cos Z\ \sin\left(\frac{Z}{2}\right)\right\}$$

$$+\sum_{n=2}^{N} X(n)\left\{\sin(nZ)\cos Z\ \sin\left(\frac{Z}{2}\right) - \cos(nZ)\sin Z\ \sin\left(\frac{Z}{2}\right)\right\}$$

$$= X(1)\sin\left(\frac{Z}{2}\right)\sin Z + \sum_{n=2}^{N} X(n)\sin(nZ)\sin\left(\frac{Z}{2}\right) + \sum_{n=2}^{N} X(n)\sin[(n-1)Z]\sin\left(\frac{Z}{2}\right)$$

$$= \sum_{n=1}^{N} X(n)\sin(nZ)\sin\left(\frac{Z}{2}\right) + \sum_{n=2}^{N} X(n)\sin[(n-1)Z]\sin\left(\frac{Z}{2}\right) \qquad (9)$$

Using (3), the above expression can be written as

$$Y(k)\sin Z = a_0 \sin Z \sin\left(\frac{Z}{2}\right) + a_1 \sin Z \sin\left(\frac{Z}{2}\right)$$

Hence,

$$Y(k) = (a_0 + a_1)\sin\left(\frac{Z}{2}\right)$$

$$= (a_0 + a_1)\sin\left(\frac{\pi k}{2N}\right) \qquad (10)$$

The a_r can be generated recursively from the input sequence $X(n)$ according to (8). Then the *k*th component of 1-D DST can be realized by (10).

3. EXAMPLE FOR REALIZING 1-D DST

Let us use a 5-point DST with input sequence $\{X(n): n = 1, 2, 3, 4, 5\}$ to clarify the proposal.

As $a_5 = 0$ and $a_6 = 0$, we get from (8)

$$\begin{aligned} a_0 &= X(1) + 2a_1 \cos Z - a_2 \\ a_1 &= X(2) + 2a_2 \cos Z - a_3 \\ a_2 &= X(3) + 2a_3 \cos Z - X(5) \\ a_3 &= X(4) + 2X(5)\cos Z \\ a_4 &= X(5) \end{aligned} \qquad (11)$$

where $Z = \dfrac{\pi k}{5}$ for $N = 5$

Figure 1 shows the block diagram of the recursive filter structure with the input sequence in reverse order for realization of 5-point 1-D DST given by (10). The values of a_0 and a_1 are implemented by (11). The delay elements Z^{-1} of the recursive filter must be reset to zero before recursion.

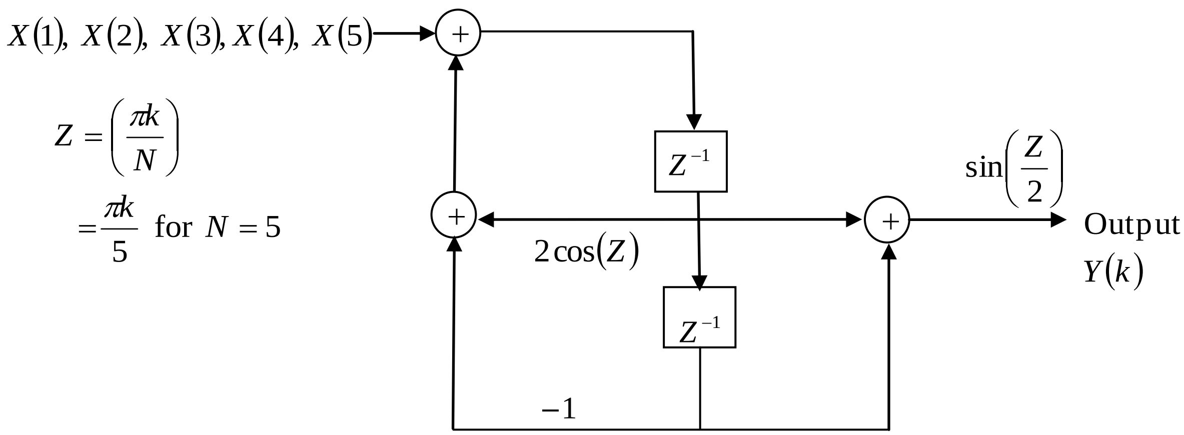


Figure 1. *Recursive filter structure for computing 5-point 1-D DST.*

4. COMPARISON OF THE PROPOSED REALIZATION FOR DST WITH EXISTING SYSTEMS

The proposed structure requires N multiplications and $(2N-2)$ additions for realization of N length 1-D DST. In Tables 1 and 2, the number of multipliers and the number of adders in the proposed algorithm are compared with the corresponding parameters based on the other methods. Table 3 gives the comparison of the computation complexities of the proposed algorithm with other algorithms.

Table 1. *Comparison of the number of multipliers required by different algorithms to compute the N-point DST*

N	**[16, 17, 19]**	**[18]**	**[12]**	**[20]**	**Proposed**
4	4	11	2	5	4
8	12	19	8	13	8
16	32	36	30	29	16
32	80	68	54	61	32
64	192	132	130	125	64

Table 2. *Comparison of the number of adders required by different algorithms to compute the N-point DST*

N	**[16, 17, 19]**	**[12]**	**[18]**	**[20]**	**Proposed**
4	9	4	11	14	6
8	29	22	26	26	14
16	81	62	58	50	30
32	209	166	122	98	62
64	513	422	250	194	126

Table 3. *Comparison of the computation complexities*

Algorithm	No. of multiplications	No. of additions
Proposed algorithm	N	$2N$-2
[13, 14, 17, 19]	(1/2) $N \log_2 N$	(3/2) $N \log_2 N - N + 1$
[15]	(1/2) $N \log_2 N$ + (1/4) N-1	(3/2) $N \log_2 N$ + (1/2) N-2
[18]	$2(N+3)(N-1) / N$	$2(2N-1)(N-1) / N$
[20] if N is even	$2N$-3	$3N$+2
[20] if N is odd	$2(N-1)$	$3N$+4

It can be seen that the proposed algorithm for DST requires less number of additions and multiplications in comparison with other architectures [12] - [20].

5. SYSTOLIC ARCHITECTURE FOR REALIZATION OF DST

The structure of the proposed linear systolic array for computation of N-point DST is shown in Fig. 2. It consists of (N+1) locally connected processing elements (PEs) of which the first N PEs are identical. The recurrence relation given by (8) is implemented in the first N PEs, while the last PE computes the DST components given by (10). Function of each of the first N PEs is shown in Fig. 3 and that of the last PE is shown in Fig. 4. One sample of the input data is fed to each PE, one time-step staggered with respect to the input of previous PE, i.e, ith input sample is fed to (N+1-i) th PE in (N+1-i) th time-step. The first output for k = 1 is obtained after (N+1) time steps and the rest (N-1) outputs for k = 2, 3, …., N are obtained in subsequent time-steps. Successive sets of N-point DSTs are obtained in every N+1 time-steps. Each PE of the linear array comprises of one multiplier and two adders, while the last PE contains one adder and one multiplier. The duration of the cycle period is $T = T_M + 2T_A$, where T_M and T_A are, respectively, the times involved in performing one multiplication and one addition in the PE. This architecture requires N multiplications per point and (2N-2) additions per point for realisation of N-point DST. The hardware - and time-complexities of the proposed systolic realisation along with those of the existing structures [21] - [23] are listed in Table 4.

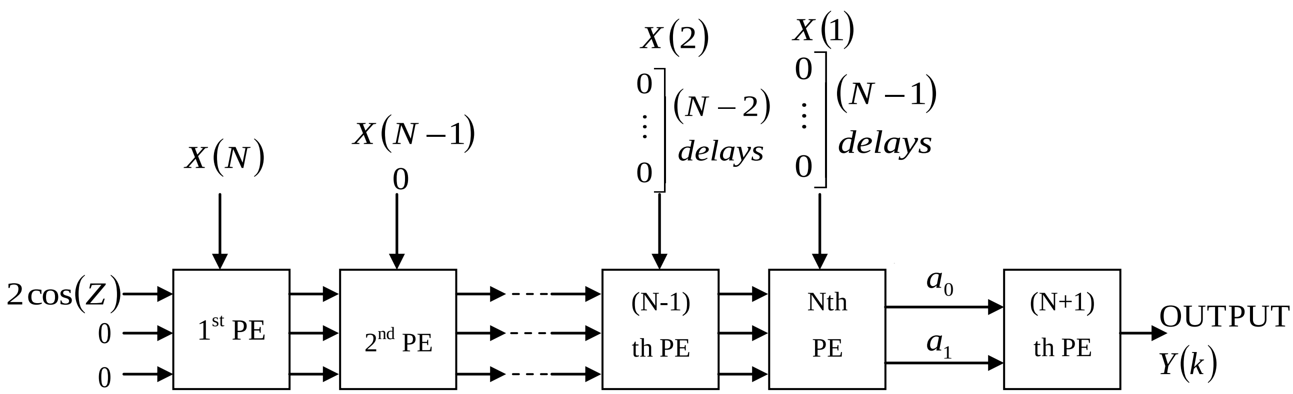
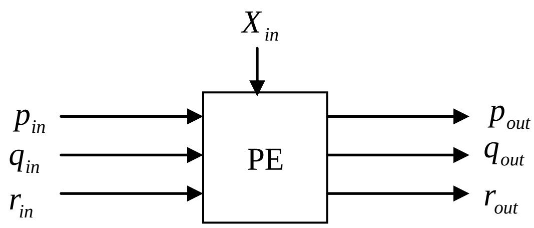


Figure 2. *The linear systolic array for N- Point DST*

$$2\cos(Z) = 2\cos\left(\frac{\pi k}{N}\right)$$

k = 1 for first (N+1) time steps.
Then $k \rightarrow k + 1$ in each time-step.



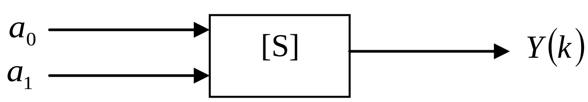
$p_{out} = p_{in}$

$q_{out} = X_{in} + p_{in}q_{in} - r_{in}$

$r_{out} = q_{in}$

X_{in} = Input sample.

Figure 3. *Function of each of the first N PEs of the linear array*



$Y(k) = S\ [a_0 + a_1]$

$$S = \sin\left(\frac{\pi k}{2N}\right)$$

k = 1 for first (N+1) time - steps. Then $k \rightarrow k$+1 in each time-step.

Figure 4. *Function of (N+1)th PE of the linear array.*

Table 4. *Hardware - and time-complexities of proposed structure and the existing systolic structures for the DST*

Structures	Multipliers	Adders	Cycle-Time (T)	Average Computation - Time
Pan and Park [21]	N	2N	$T_M + T_A$	NT/2
Fang and Wu [22]	N/2 + 3	N + 3	$T_M + 2T_A$	NT
Chiper *et al*. [23]	N-1	N+1	$T_M + T_A$	(N-1) T/2
Proposed	N	2N - 2	$T_M + 2T_A$	(N+1) T

6. PROPOSED RECURSIVE ALGORITHM FOR IDST

The inverse discrete sine transform (IDST) of data $\{Y(k) : k = 1,2,\ldots, N\}$ is given by

$$X(n) = \sum_{k=1}^{N} Y(k) \sin\left[\frac{k(2n-1)\pi}{2N}\right] \qquad \text{for } n = 1, 2, \ldots, N \tag{12}$$

A recursive kernel b_m for IDST is introduced as given below

$$b_m \sin\alpha = \sum_{k=m+1}^{N} Y(k) \sin[(k-m)\alpha] \tag{13}$$

Following the same procedure for deriving (7) from (3), we get

$$b_m \sin\alpha = Y(m+1)\sin\alpha + 2b_{m+1}\sin\alpha\cos\alpha - b_{m+2}\sin\alpha \tag{14}$$

Therefore,

$$b_m = Y(m+1) + 2b_{m+1}\cos\alpha - b_{m+2} \tag{15}$$

for $m = 0, 1, 2, ..., N-1$ and $b_m = 0$ if $m \geq N$.

Let $\alpha = \frac{(2n-1)\pi}{2N}$, then (12) can be written as

$$X(n) = \sum_{k=1}^{N} Y(k)\sin(k\alpha) \tag{16}$$

For $m = 0$, the recursive kernel in (13) is given by

$$b_0 \sin\alpha = \sum_{k=1}^{N} Y(k)\sin(k\alpha) \tag{17}$$

From (16) and (17), we have

$$\begin{aligned} X(n) &= b_0 \sin\alpha \\ &= b_0 \sin\left[\frac{(2n-1)\pi}{2N}\right] \end{aligned} \tag{18}$$

for $n = 1, 2, ..., N$

The IDST can be realized using (15) and (18).

7. EXAMPLE FOR REALIZING IDST

Let us use a 5-point IDST with output data sequence $\{Y(k): k = 1, 2, 3, 4, 5\}$ to clarify the proposal. As b_5=0 and b_6 =0, we have from (15)

$$\begin{aligned} b_0 &= Y(1) + 2b_1\cos\alpha - b_2 \\ b_1 &= Y(2) + 2b_2\cos\alpha - b_3 \\ b_2 &= Y(3) + 2b_3\cos\alpha - Y(5) \\ b_3 &= Y(4) + 2Y(5)\cos\alpha \\ b_4 &= Y(5) \end{aligned} \tag{19}$$

Figure 5 shows the recursive filter structure for implementation of 5-point IDST using (18) and (19).

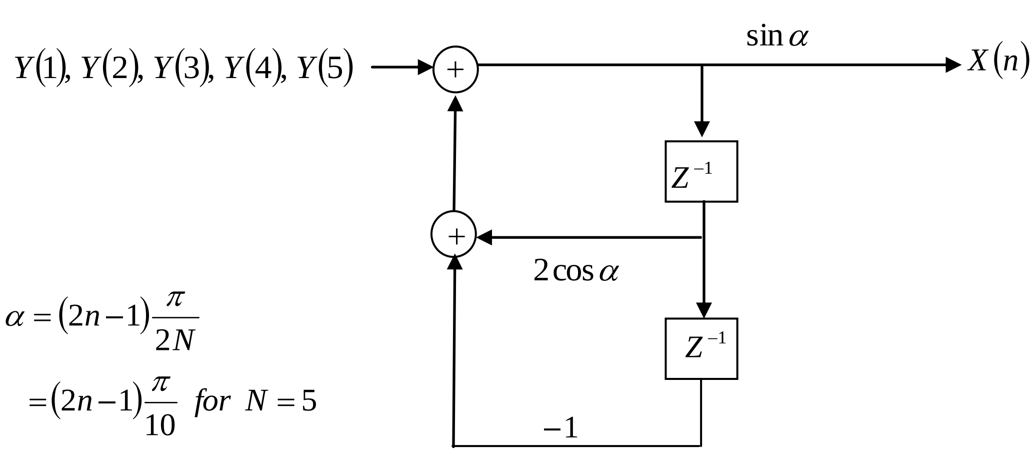


Figure 5. *Recursive filter structure for computing 5-point IDST.*

8. CONCLUSION

In this paper, two recursive algorithms for realizing DST and IDST of any length have been derived. Also a linear systolic architecture for implementing the recursive algorithm for DST is presented. The number of additions and multiplications in the algorithm for DST are less in comparison with some existing structures. Therefore, saving in time can be achieved by the proposed algorithm for DST in its realization. The recursive structures require less memory and are suitable for parallel VLSI implementation.

9. 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. Poornachandra, V. Ravichandran and N.Kumarvel, "Mapping of discrete cosine transform (DCT) and discrete sine transform (DST) based on symmetries" IETE Journal of Research, Vol. 49, no. 1, pp 35-42, January-February 2003.

[6] 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.

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

[8] 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.

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

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

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

[12] P. Yip and K.R. Rao, "A fast computational algorithm for the discrete sine transform", IEEE Trans. Commun., vol. COM-28, pp. 304-307, Feb. 1980.

[13] P. Yip and K.R. Rao, "Fast decimation-in-time algorithms for a family of discrete sine and cosine transforms", Circuits, Syst., Signal Processing, vol. 3, pp. 387-408, 1984.

[14] O. Ersoy and N.C. Hu, "A unified approach to the fast computation of all discrete trigonometric transforms," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 1843-1846, 1987.

[15] P. Yip and K.R. Rao, "The decimation-in-frequency algorithms for a family of discrete sine and cosine transforms", Circuits, Syst., Signal Processing, vol. 7, no. 1, pp. 3-19, 1988.

[16] A. Gupta and K.R. Rao, "A fast recursive algorithm for the discrete sine transform" IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 38, no. 3, pp. 553-557, March, 1990.

[17] Z. Cvetković and M.V. Popović, "New fast recursive algorithms for the computation of discrete cosine and sine transforms", IEEE Trans. Signal Processing, vol. 40, no. 8, pp. 2083-2086, Aug. 1992.

[18] J. Caranis, "A VLSI architecture for the real time computation of discrete trigonometric transform", J. VLSI Signal Process., no. 5, pp. 95-104, 1993.

[19] Peizong Lee and Fang-Yu Huang, "Restructured recursive DCT and DST algorithms", IEEE Transactions on Signal Processing," vol. 42, no. 7, pp. 1600-1609, July 1994.

[20] V. Kober, "Fast recursive algorithm for sliding discrete sine transform", Electronics Letters, vol. 38, no. 25, pp. 1747-1748, Dec. 2002.

[21] S.B. Pan and R.H. Park, "Unified systolic array for computation of DCT / DST / DHT", IEEE Trans. Circuits Syst. Video Technol., vol. 7, no. 2, pp.413-419, April 1997.

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

[23] D.F. Chiper, M.N.S. Swamy, M.O. Ahmad, and T. Stouraitis, "A systolic array architecture for the discrete sine transform", IEEE trans. Signal Process., vol. 50, no. 9, pp. 2347 - 2354, Sept. 2002.