RESEARCH ARTICLE OPEN ACCESS

# Recursive Algorithms and Systolic Architectures for Realization of Type-II Discrete Cosine Transform and Inverse Discrete Cosine Transform

M.N.Murty
(Department of Physics, National Institute of Science and Technology, Berhampur-761008, Odisha, India)

**ABSTRACT**
The paper presents novel recursive algorithms for realization of one-dimensional type-II discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) of any length. By using some mathematical techniques, recursive expressions for DCT and IDCT have been developed. The number of additions and multiplications in the recursive algorithm for DCT are less in comparison with some other DCT algorithms. Basing on these two recursive algorithms, two systolic architectures are presented for realization of DCT and IDCT. The recursive algorithms are appropriate for VLSI implementation.
***Keywords-***Discrete cosine transform, Inverse discrete cosine transform, Recursive algorithm, Systolic architecture.

## I. INTRODUCTION

Discrete transforms play a significant role in digital signal processing. Discrete cosine transform (DCT) is used as key function in many signal and image processing applications. There are four types of DCT. Of these, the DCT-II and DCT-IV have gained popularity.

The original definition of the DCT introduced by Ahmed *et al*. in 1974 [1] was one-dimensional (1-D) and suitable for 1-D digital signal processing. The DCT has found wide applications in speech and image processing as well as telecommunication signal processing for the purpose of data compression, feature extraction, image reconstruction, and filtering. Thus, many algorithms and VLSI architectures for the fast computation of DCT have been proposed [2]-[7]. Among those algorithms [6] and [7] are believed to be most efficient two-dimensional DCT algorithms in the sense of minimizing any measure of computational complexity.

In this paper, two algorithms to convert 1-D type-II DCT and IDCT of any size into recursive forms are presented. These algorithms are implemented by recursive filter structures. Two systolic architectures for realization of DCT and IDCT of arbitrary length are presented in this paper. The proposed approach requires $N$ multiplications and $(3N-4)$ additions for realization of $N$ length DCT. The number of multiplications and additions in the proposed algorithm for DCT are less in comparison with some existing structures. The IDCT requires $N$ multiplications and $(2N - 3)$ additions for its realization.

The systolic architecture has the following characteristics:

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

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 VLSI chips. The highly regular structure of systolic circuits renders them comparatively easy to design and test. The systolic arrays are used in the design and

implementation of high performance digital signal processing equipment.

The rest of the paper is organized as follows: The derivation of recursive algorithm for 1-D DCT-II is presented in Section-II. An example for realization of DCT is given in Section-III. The comparison of proposed realization of DCT with other research works is presented in Section-IV. The systolic architecture for computation DCT is presented in Section-V. The recursive algorithm for IDCT is given in Section-VI. An example for realization IDCT is presented in Section-VII. The systolic architecture for realization of IDCT is presented in Section-VIII. The conclusion is given in Section-IX.

## II. PROPOSED RECURSIVE ALGORITHM FOR DCT-II

The type-II DCT of input sequence $\{x(n): n = 0,1,..., N-1\}$ is defined as

$$Y(k) = \varepsilon(k)\sqrt{\frac{2}{N}}\sum_{n=0}^{N-1} x(n)\cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (1)$$

for $k$ = 0, 1, 2,…., $N$-1

where,

$$\varepsilon(k) = \begin{cases} \frac{1}{\sqrt{2}} if\ k = 0 \\ 1 if\ k = 1,2,....., N-1 \end{cases} \quad (2)$$

The $Y$ values represent the output data. Without loss of generality, the scale factor $\varepsilon(k)\sqrt{\frac{2}{N}}$ can be omitted in rest of the paper.

Replacing $n$ by ($N$ - $n$) in (1), we obtain

$$Y(k) = (-)^k \sum_{n=1}^{N} x(N-n)\cos\left[\left(n-\frac{1}{2}\right)\frac{k\pi}{N}\right] \quad (3)$$

for $k$ = 0, 1, 2,…, $N$-1

Define

$$T_n^k = \frac{\cos\left[\left(n-\frac{1}{2}\right)\theta_k\right]}{\cos\left(\frac{\theta_k}{2}\right)} \quad (4)$$

where, $\theta_k = \frac{k\pi}{N}$

Using (4), (3) can be expressed as

$$Y(k) = (-)^k \cos\left(\frac{\theta_k}{2}\right)\sum_{n=1}^{N} x(N-n)T_n^k \quad (5)$$

for $k$ = 0,1,2,.., $N$ -1

From (4), we get

$$T_{n+1}^k = \frac{\cos\left[\left(n+\frac{1}{2}\right)\theta_k\right]}{\cos\left(\frac{\theta_k}{2}\right)} \quad (6)$$

and

$$T_{n-1}^k = \frac{\cos\left[\left(n-\frac{3}{2}\right)\theta_k\right]}{\cos\left(\frac{\theta_k}{2}\right)} \quad (7)$$

It is easy to show the following trigonometric identity.

$$\cos(r\theta_k) = 2\cos[(r-1)\theta_k]\cos(\theta_k) - \cos[(r-2)\theta_k] \quad (8)$$

Using (8) in the numerator of RHS of (6), we have

$$T_{n+1}^k = \frac{2\cos\left[\left(n-\frac{1}{2}\right)\theta_k\right]\cos(\theta_k) - \cos\left[\left(n-\frac{3}{2}\right)\theta_k\right]}{\cos\left(\frac{\theta_k}{2}\right)} \quad (9)$$

Using (4) and (7) in (9), we obtain the following recursive relation.

$$T_{n+1}^k = T_n^k 2\cos(\theta_k) - T_{n-1}^k \quad (10)$$

Define

$$P_N^k = \sum_{n=1}^{N} x(N-n) T_n^k \quad (11)$$

As $T_1^k = 1$ from (4), (11) can be expressed as

$$\begin{aligned} P_N^k &= x(N-1) + \sum_{n=2}^{N} x(N-n) T_n^k \\ &= x(N-1) + \sum_{n=1}^{N-1} x(N-n-1) T_{n+1}^k \end{aligned} \quad (12)$$

Using the recursive formula (10) in (12), we obtain

$$P_N^k = x(N-1) + \sum_{n=1}^{N-1} x(N-n-1)\left[T_n^k 2\cos(\theta_k) - T_{n-1}^k\right]$$

$$= x(N-1) + 2\cos(\theta_k)\sum_{n=1}^{N-1} x(N-n-1) T_n^k - \sum_{n=1}^{N-1} x(N-n-1) T_{n-1}^k$$

$$= x(N-1) + 2\cos(\theta_k)\sum_{n=1}^{N-1} x(N-n-1) T_n^k - x(N-2) T_0^k - \sum_{n=2}^{N-1} x(N-n-1) T_{n-1}^k$$

As $T_0^k = 1$ from (4), the above expression becomes

$$P_N^k = x(N-1) + 2\cos\theta_k \sum_{n=1}^{N-1} x(N-n-1) T_n^k - x(N-2) - \sum_{n=2}^{N-1} x(N-n-1) T_{n-1}^k \quad (13)$$

Form (11), we get

$$P_{N-1}^k = \sum_{n=1}^{N-1} x(N-1-n)\, T_n^k \quad (14)$$

and

$$\begin{aligned} P_{n-2}^K &= \sum_{N=1}^{n-2} x(N-2-n) T_n^k \\ &= \sum_{n=2}^{N-1} x(N-1-n) T_{n-1}^k \end{aligned} \quad (15)$$

Using (14) and (15) in (13), we obtain the following recursive relation.

$$P_N^k = x(N-1) - x(N-2) + 2\cos\theta_k P_{N-1}^k - P_{N-2}^k \quad (16)$$

Substituting (11) in (5), we get

$$Y(k) = (-1)^k \cos\left(\frac{\theta_k}{2}\right) P_N^k \quad (17)$$

for $k$ = 0,1,2,…,$N$-1

The DCT-II in (17) can be realized using the recursive formula (16).

## III. EXAMPLE FOR REALIZING DCT-II

A 5-point DCT-II with input sequence $\{x(n) : n = 0,1,2,3,4\}$ is taken to clarify the proposal. For $N$ =5, (17) can be written as

$$Y(k) = (-1)^k \cos\left(\frac{\theta_k}{2}\right) P_5^k \quad (18)$$

for $k$ = 0,1,2,3,4.

As $P_0^k = 0, P_{-1}^k = 0$ and $x(-1) = 0$, we get the following relations from (16).

$$\begin{aligned} P_1^k &= x(0) \\ P_2^k &= [x(1) - x(0)] + 2\cos\theta_k\, x(0) \\ P_3^k &= [x(2) - x(1)] + 2\cos\theta_k\, P_2^k - x(0) \\ P_4^k &= [x(3) - x(2)] + 2\cos\theta_k\, P_3^k - P_2^k \\ P_5^k &= [x(4) - x(3)] + 2\cos\theta_k\, P_4^k - P_3^k \end{aligned} \quad (19)$$

The DCT-II given in (17) can be realized using the recursive formula (16) by the recursive filter structure, shown in Fig. 1. $Z^{-1}$ represents unit-delay element.

$$(-1)^k \cos\left(\frac{\theta_k}{2}\right)$$

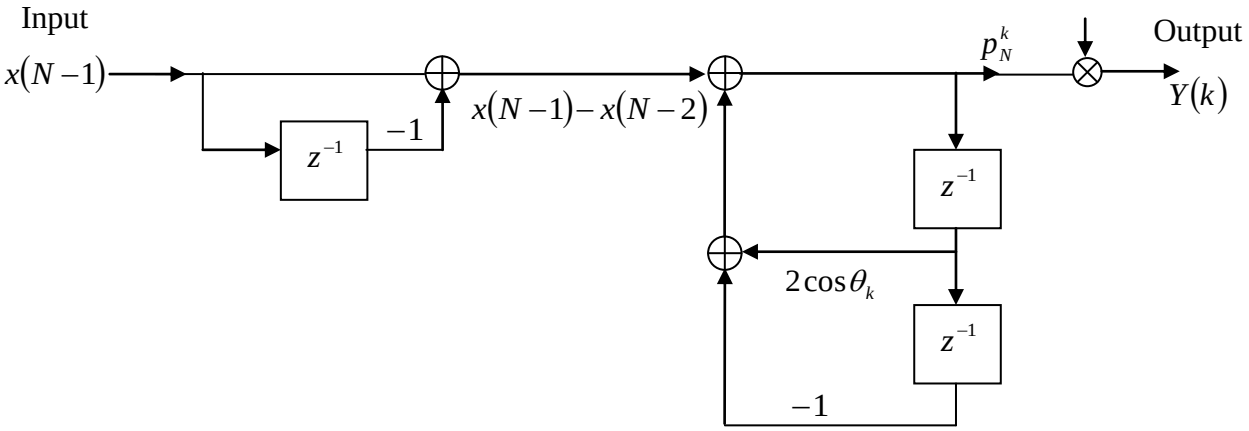



**Fig. 1: Recursive filter structure for computing DCT-11**

## IV. COMPARISON OF DCT-II WITH RELATED WORKS

The proposed approach requires $N$ multiplications and (3$N$-4) additions for the realization of $N$ length DCT-II. 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 found in the related research works.

**Table 1: Comparison of the number of multipliers required by different algorithms**

| $N$ | [8] | [4] | [11] | [5,13] | [14] | [12] | Proposed |
|---|---|---|---|---|---|---|---|
| 4 | 6 | 5 | 5 | 4 | 11 | 4 | 4 |
| 8 | 16 | 17 | 13 | 12 | 19 | 13 | 8 |
| 16 | 44 | 49 | 33 | 32 | 36 | 35 | 16 |
| 32 | 116 | 129 | 81 | 80 | 68 | 87 | 32 |
| 64 | 292 | 321 | 193 | 192 | 132 | 207 | 64 |

**Table 2: Comparison of the number of adders required by different algorithms**

| $N$ | [11] | [4] | [5,13] | [8] | [14] | [12] | Proposed |
|---|---|---|---|---|---|---|---|
| 4 | 9 | 9 | 9 | 8 | 11 | 12 | 8 |
| 8 | 35 | 41 | 29 | 26 | 26 | 38 | 20 |
| 16 | 95 | 129 | 81 | 74 | 58 | 102 | 44 |
| 32 | 251 | 353 | 209 | 194 | 122 | 254 | 92 |
| 64 | 615 | 897 | 513 | 482 | 250 | 606 | 188 |

**Table 3: Computation complexities**

| | of multiplications | of additions |
|---|---|---|
| Proposed algorithm | $N$ | 3$N$-4 |
| F[5,9,10,13] | (1/2) $N\log_2 N$ | (3/2) $N\log_2 N$ - $N$ + 1 |
| [4,15,16] | $N\log_2 N$ /2 + 1 | 3 $N\log_2 N$ / 2 -$N$ +1 |
| [12] | (1/2) $N\log_2 N$ + (1/4) $N$-1 | (3/2) $N\log_2 N$ + (1/2) $N$-2 |
| [14] | 2($N$+3)($N$-1) / $N$ | 2(2$N$-1)($N$-1) / $N$ |

## V. SYSTOLIC ARCHITECTURE FOR DCT-II

The structure of the proposed linear systolic array for computation of $N$-point DCT-II is shown in Fig. 2. The DCT is implemented by this systolic architecture as per the recursive equations (19) given for $N$=5. It consists of ($N$+1) locally connected processing elements (PEs) of which the first $N$ PEs are identical. The recurrence relation given by (16) is implemented in the first $N$ PEs, while the last PE computes the DCT components given by (17). The function of a subtractor cell is shown in Fig.3. The function of each of the first $N$ PEs is shown in Fig. 4 and that of the last PE is shown in Fig. 5. One sample of the input data is fed to each PE through a subtractor cell one time-step staggered with respect to the input of previous PE. The black solid circle, "•" represents one-cycle delay element. The input to the $i$th PE of the first $N$ PEs is [$x(i$-1) - $x(i$ - 2)] due to the delay element between its sutractor cell and the data sample fed to the previous PE. The first output is obtained after ($N$+1) time steps and the rest ($N$ – 1) output are obtained in the subsequent ($N$ – 1) time steps. Each PE of the linear array consists of one multiplier and two adders, while the last ($N$ + 1)th PE contains one multiplier. The duration of the cycle period is $T = T_M + 3T_A$, where $T_M$ and $T_A$ are, respectively, the times involved in performing one multiplication and one addition. This architecture requires $N$ multiplications and (3$N$ – 4) additions for realization of $N$-point DCT.

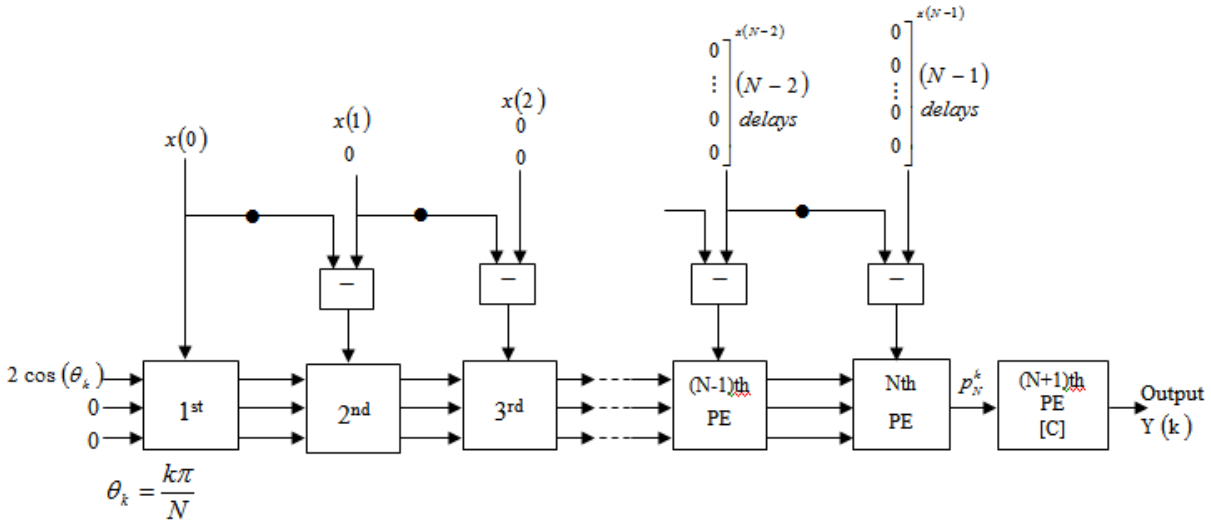



**Fig. 2: The linear systolic architecture for *N*-point DCT-II**

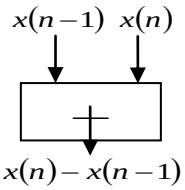
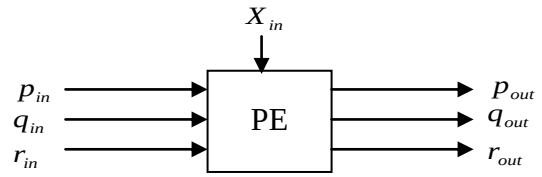
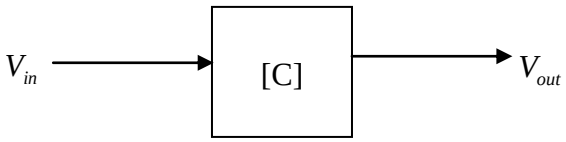



**Fig.3: Subtractor cell of the linear systolic architecture for DCT-II**




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

**Fig.4: Function of each of the first *N* PEs of systolic architecture for DCT-II**




$$V_{out} = V_{in} C$$
$$C = (-1)^k \cos\left(\frac{\theta_k}{2}\right)$$

**Fig.5: Function of (*N*+1)th PE of the linear systolic architecture for DCT-II**

## VI. PROPOSED RECURSIVE ALGORITHM FOR IDCT

The IDCT of a sequence $\{Y(k): k = 0,1,2,...,N-1\}$ can be written as

$$x(n) = \sum_{k=0}^{N-1} Y(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (20)$$

for $n = 0,1,2,...,N-1$

Replacing $k$ by $(N - k)$ in (20), we obtain

$$x(n) = (-1)^n \sum_{k=1}^{N} Y(N-k) \sin\left[\left(n+\frac{1}{2}\right)\frac{k\pi}{N}\right] \quad (21)$$

Taking $\theta_n = \left(n+\frac{1}{2}\right)\frac{\pi}{N}$, (21) can be written as

$$x(n) = (-1)^n \sum_{k=1}^{N} Y(N-k) \sin(k\theta_n) \quad (22)$$

Define

$$R_k^n = \frac{\sin(k\theta_n)}{\sin\theta_n} \quad (23)$$

Using (23) in (22), we obtain

$$x(n) = (-1)^n \sin\theta_n \sum_{k=1}^{N} Y(N-k) R_k^n \quad (24)$$

for $n = 0,1,2,...,N-1$

From (23), we have

$$R_{k+1}^n = \frac{\sin(k+1)\theta_n}{\sin\theta_n} \quad (25)$$

$$R_{k-1}^n = \frac{\sin(k-1)\theta_n}{\sin\theta_n} \quad (26)$$

Consider the following trigonometric identity

$$\sin(r\theta_n) = 2\sin[(r-1)\theta_n]\cos\theta_n - \sin[(r-2)\theta_n] \quad (27)$$

Using (27) in (25), we get

$$R_{k+1}^n = \frac{2\sin(k\theta_n)\cos\theta_n - \sin[(k-1)\theta_n]}{\sin\theta_n} \quad (28)$$

Using (25) and (26) in (28), we obtain the following recursive formula.

$$R_{k+1}^n = 2\cos(\theta_n)R_k^n - R_{k-1}^n \quad (29)$$

Define

$$Q_N^n = \sum_{k=1}^{N} Y(N-k)R_k^n \quad (30)$$

Since $R_1^n = 1$ from (23), the above expression (30) can be written as

$$Q_N^n = Y(N-1) + \sum_{k=2}^{N} Y(N-k)R_k^n$$

$$= Y(N-1) + \sum_{k=1}^{N-1} Y(N-k-1)R_{k+1}^n \quad (31)$$

Using the recursive relation (29), (31) can be expressed as

$$Q_N^n = Y(N-1) + \sum_{k=1}^{N-1} Y(N-k-1)\left[2\cos(\theta_n)R_k^n - R_{k-1}^n\right]$$

$$= Y(N-1) + 2\cos\theta_n \sum_{k=1}^{N-1} Y(N-k-1)R_k^n - \sum_{k=1}^{N-1} Y(N-k-1)R_{k-1}^n$$

$$= Y(N-1) + 2\cos\theta_n \sum_{k=1}^{N-1} Y(N-k-1)R_k^n - \sum_{k=0}^{N-2} Y(N-k-2)R_k^n \quad (32)$$

Since $R_0^n = 0$ from (23), the above expression can be written as

$$Q_N^n = Y(N-1) + 2\cos\theta_n \sum_{k=1}^{N-1} Y(N-k-1)R_k^n - \sum_{k=1}^{N-2} Y(N-k-2)R_k^n \quad (33)$$

Using (30), we get

$$Q_{N-1}^n = \sum_{k=1}^{N-1} Y(N-k-1)R_k^n \quad (34)$$

and

$$Q_{N-2}^n = \sum_{k=1}^{N-2} Y(N-k-2)R_k^n \quad (35)$$

Substituting (34) and (35) in (33), the following recursive relation is obtained.

$$Q_N^n = Y(N-1) + 2\cos(\theta_n)Q_{N-1}^n - Q_{N-2}^n \quad (36)$$

Using (30) in (24), we have

$$x(n) = (-1)^n \sin(\theta_n) Q_N^n \quad (37)$$

for $n$ = 0,1,2,…..,$N$-1

The IDCT in (37) can be computed using the recursive relation (36). This algorithm requires (2$N$-3) additions and $N$ multiplications for realization of IDCT.

## VII. EXAMPLE FOR REALIZING IDCT

Let us consider a 5-point IDCT with output sequence $\{Y(k): k = 0,1,2,3,4\}$ to clarify the proposal. For $N$ =5, (37) can be written as

$$x(n) = (-1)^n \sin(\theta_n) Q_5^n \quad (38)$$

for $n$ = 0,1,2,3,4.

As $Q_0^n = 0$ and $Q_{-1}^n = 0$, we get the following recursive relations from (36)

$$\begin{aligned} Q_1^n &= Y(0) \\ Q_2^n &= Y(1) + 2\cos(\theta_n) Y(0) \\ Q_3^n &= Y(2) + 2\cos(\theta_n) Q_2^n - Y(0) \\ Q_4^n &= Y(3) + 2\cos(\theta_n) Q_3^n - Q_2^n \\ Q_5^n &= Y(4) + 2\cos(\theta_n) Q_4^n - Q_3^n \end{aligned} \quad (39)$$

The IDCT given in (37) can be computed using the recursive relation (36) by the recursive filter structure, shown in Fig. 6.

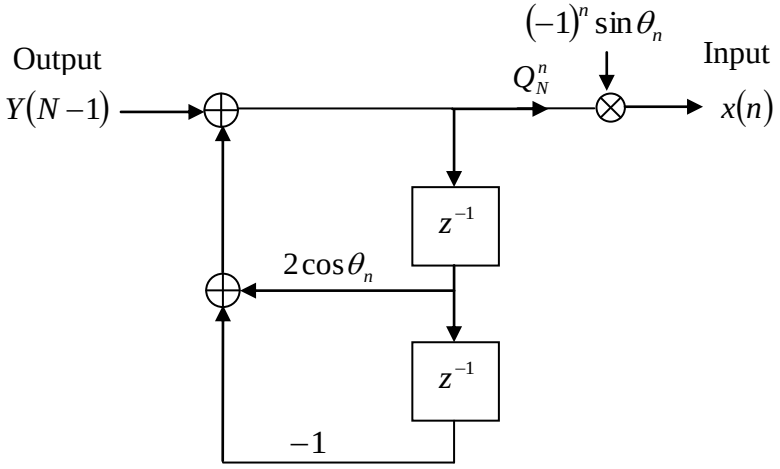
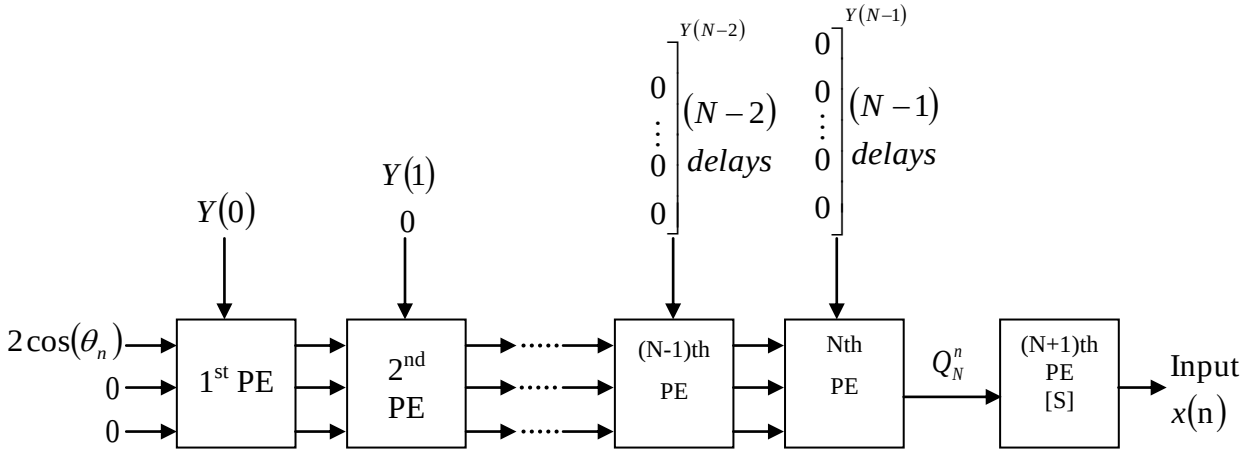



**Fig.6: Recursive filter structure for computing IDCT-11**

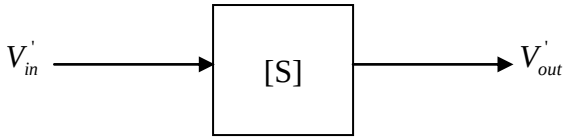
## VIII. SYSTOLIC ARCHITECTURE FOR IDCT-II

The structure of the proposed linear systolic architecture for computation of *N*-point IDCT is shown in Fig. 7. The IDCT is realized by this systolic architecture as per the recursive equations (39) given for *N* = 5. It consists of (*N* + 1) locally connected PEs of which the first *N* PEs are identical. The recurrence relation given by (36) is implemented in the first *N* PEs, while the last PE computes the IDCT components given by (37). The function of each of the first *N* PEs is shown in Fig. 4 and that of the last PE is shown in Fig. 8. The output data is fed to each PE one time-step staggered with respect to the input of previous PE. The input to the *i*th PE of the first *N* PEs is *Y*(*i*-1). The first input data is obtained after (*N*+1) time steps and the rest (*N* – 1) input data are obtained in the subsequent (*N* – 1) time steps. Each PE of the linear array consists of one multiplier and two adders, while the last (*N* + 1)th PE contains 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. The average computation time is (*N* + 1)*T*. This architecture requires *N* multiplications and (2*N* – 3) additions for realization of *N*-point IDCT.




$$\theta_n = \left(n + \frac{1}{2}\right)\frac{\pi}{N}$$

**Fig.7: The linear systolic architecture for *N*-point IDCT**




$$V'_{out} = V'_{in} S$$
$$S = (-)^n \sin\theta_n$$

**Fig.8: Function of *(N*+1)th PE of the linear systolic architecture for IDCT**

## IX. CONCLUSION

In this paper, two novel recursive algorithms for realizing one-dimensional type-II DCT and IDCT of any length have been derived. These algorithms are implemented by recursive filter structures. Also two linear systolic architecture for realizing DCT and IDCT are presented in this paper. The number of additions and multiplications in the recursive algorithm for DCT are less in comparison with some existing structures. Therefore, saving in time can be achieved by the proposed algorithm for DCT in its realization. The recursive structures require less memory and are suitable for parallel VLSI implementation. The systolic arrays are used in the design and implementation of high performance digital signal processing equipment. The highly regular structure of systolic circuits renders them comparatively easy to design and test.

## REFERENCES

[1] N.Ahmed, T.Natarajan, and K.R.Rao,Discrete cosine transform, *IEEE Trans. Comput.*, vol. C-23, 1974,90-93.

[2] R.K.Rao and P.Yip, *Discrete cosine transform: algorithm, advantages, and applications*,(New York: Academic, 1990).

[3] M.J.Narasimha and A.M.Peterson, On the computation of the discrete cosine transform, *IEEE Trans. Communications*, vol.COM-26, no.6, June 1978,934-936.

[4]. H.S.Hou, A fast recursive algorithms for computing the discrete cosine transform, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35,no.10, Oct.1987,1455-1461.

[5] P.Lee and F.Y.Huang, Reconstructed recursive DCT and DST algorithms, *IEEE Trans.SignalProcessing*,vol.42,no.7,Jul.1994, 1600- 1609.

[6] P.Duhamel and C.Guillemot, Polynomial transform computation of 2-D DCT, *in Proc. ICASSP'90*, Apr.1990,1515-1518.

[7] E.Feig and S.Winograd, Fast algorithms for the discrete cosine transform, *IEEE Trans. Signal Processing*, vol.40, no.9, Sep.1992, 2174- 2193.
[8] W.H Chen, C.H.Smith and S.C.Fralick, A fast computational algorithm for the discrete cosine transform, *IEEE Trans. Communicat*., vol. COM-25, no. 9, Sep.1977, 1004-1009.
[9] 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, 1984, 387-408.
[10] 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*, 1987, 1843-1846.
[11] H.S.Malvar, Corrections to fast computation of the discrete cosine transform and the discrete hartley transform, *IEEE Trans Acoust., Speech, Signal Processing*, vol. 36, no. 4, April 1988, 610-612.
[12] 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, 1988, 3-19.
[13] Z.Cvetković and M.V.Popović, New fast recursive algorithms for the computation of discrete cosine and sine transforms, *IEEE Trans.SignalProcessing*, vol.40, no.8,August 1992, 2083-2086.
[14] J.Caranis, A VLSI architecture for the real time computation of discrete trigonometric transform, *J. VLSI Signal Process*., no. 5, 1993, 95-104.
[15] V.Britanak, On the discrete cosine computation, *Signal Process*. , vol. 40, no. 2- 3, 1994, 183-194.
[16] C.W.Kok, Fast algorithm for computing discrete cosine transform, *IEEE Trans. Signal Process*., vol. 45, March 1997, 757-760.