

# Novel Formulation and Realisation of Discrete Cosine Transform using Distributed Arithmetic

Yuk-Hee Chan and Wan-Chi Siu

Department of Electronic Engineering  
Hong Kong Polytechnic  
Hung Hom, Kowloon  
Hong Kong

## Abstract

In this paper, a new algorithm is introduced such that we can convert an odd prime length, say  $N$ , DCT into two  $(N-1)/2$  length cyclic correlations. This formulation enables us to realise the DCT using the distributed arithmetic and it also results in extremely regular structure which is most suitable for VLSI realisation. An example is given to show the feasibility and the structural regularity of the algorithm.

## 1. INTRODUCTION

The discrete cosine transform (DCT) is an important tool for digital signal processing as it performs much like the optimal Karhunen-Loeve transform (KLT) under a variety of criteria. Many algorithms for the computation of the DCT have been proposed since the introduction of the DCT in 1974 by Ahmed, Natarajan and Rao[1]. Among these algorithms, three algorithms[2-4] meet the minimum known number of multiplications for their realisation. However, all of these algorithms still rely heavily on the use of multipliers which are expensive to be implemented using VLSI or other techniques.

The distributed arithmetic[5-9] is essentially a technique which is extremely regular and structural in nature such that it is most suitable for VLSI realisation. The advantages of this type of structure include: (1) no actual multiplication involved as multipliers are replaced by memory look-up tables, (2) high accuracy as it suffers fewer rounding/truncation errors than the cases for other structures, (3) possible for modular circuit design as the structure is extremely regular and (4) simple structure which leads to a saving of gate count and makes routing easy. These features allow a high speed circuit design composed of memories, adders and registers only. Besides, due to the nature of the structure of the distributed arithmetic, it is most suitable for the realisation of convolutions/correlations. It can reduce the computation of an  $N$ -length cyclic convolution from requiring  $N^2$  multiplications and  $N(N-1)$  additions to  $N(M-1)$  additions only, where  $M$  is the wordlength of the input sequence  $\{x(n):n=0,1..N-1\}$ . Hence, it is also true that the success of using distributed arithmetic for the realisation of a certain transform relies

heavily on whether we could be able to convert the transform into a convolution/correlation form. Successful results[7-9] for the realisation of discrete Fourier transforms have been reported. However, there are less significant results reported in the literature for the realisation of DCT using the distributed arithmetic. Recently, we proposed a new approach to realise a  $2^m$ -length DCT by correlations[10]. In such case, one can make use of the distributed arithmetic technique to achieve an easy and fast chip implementation.

It is found that most of fast discrete cosine transform algorithms were proposed for the computation of  $2^m$  length DCT. In this paper, a new algorithm is proposed such that we can convert an odd prime length, say  $N$ , DCT into two  $(N-1)/2$  length cyclic correlations. This formulation enables us to realise the DCT using the distributed arithmetic with the minimum number of operations and it also results in an extremely regular structure which is most suitable for VLSI realisation. An example will also be given to show the feasibility and the structural regularity of the present algorithm.

## 2. ALGORITHM DERIVATION:

The DCT[1] of a real data sequence  $\{y(i):i=0,1,..N-1\}$  is defined as

$$Y(k) = \sum_{i=0}^{N-1} y(i) \cos [2\pi(2i+1)k/4N] \quad \text{for } k=0,1,..N-1 \quad (1)$$

We firstly define another sequence  $\{x(i):i=0,1,..N-1\}$  as

$$\begin{aligned} x(N-1) &= y(N-1) \\ x(i) &= y(i) - x(i+1) \quad \text{for } i=0,1,..N-2 \quad (2) \end{aligned}$$

Then we have

$$Y(k) = \{ 2 T(k) + x(0) \} \cos[k\pi/2N] \quad \text{for } k=0,1,..N-1 \quad (3)$$

where  $T(k)$  is defined as

$$T(k) = \sum_{i=1}^{N-1} x(i) \cos[\pi ik/N] \quad \text{for } k=1,2,..N \quad (4)$$

Among  $Y(k)$ 's,  $Y(0) = \sum_{i=0}^{N-1} y(i)$ , which requires

simple additions only. For other terms of  $Y(k)$ , we have to obtain the sequence  $\{T(k); k=1, 2, \dots, N-1\}$ .

Now let  $N$  be an odd prime. If we split  $T(k)$  into odd and even sequences, we have

$$T(2k) = \sum_{i=1}^{N-1} x(i) \cos [2\pi i k / N] \quad \text{for } k=1, 2, \dots, (N-1)/2 \quad (5)$$

and

$$T(2k' + 1) = \sum_{i=1}^{N-1} x(i) \cos [\pi i (2k' + 1) / N] \quad \text{for } k'=0, 1, \dots, (N-3)/2 \quad (6)$$

Let  $2k' + 1 = N - 2k$ , then we have

$$T(N-2k) = \sum_{i=1}^{N-1} e(i) \cos [2\pi i k / N] \quad \text{for } k=1, 2, \dots, (N-1)/2 \quad (7)$$

where  $e(i) = (-1)^i x(i)$  for  $i=1, 2, \dots, (N-1)$  (8)

As  $N$  is prime, there exists a bijective mapping on the set  $\text{INDEX} = \{i; i=1, 2, \dots, N-1\}$  to itself:

$$\langle g^v \rangle_N = u \quad (9)$$

where  $u, v \in \{1, 2, \dots, N-1\}$ ,  $g$  is a primitive root of  $N$  and  $\langle x \rangle_N$  means the residue of the number  $x$  modulo  $N$ . If we extend the domain of  $T(N-2k)$  and  $T(2k)$  in eqn.(7) and (5) to  $\{k; k \in \text{INDEX}\}$ , we have

$$T'(k) = \sum_{i=1}^{N-1} x'(i) \cos [(2\pi/N) \langle g^{i+k} \rangle_N] \quad (10)$$

$$\text{and } T''(k) = \sum_{i=1}^{N-1} e'(i) \cos [(2\pi/N) \langle g^{i+k} \rangle_N] \quad (11)$$

$$\begin{aligned} \text{where } x'(i) &= x(\langle g^i \rangle_N) \\ e'(i) &= e(\langle g^i \rangle_N) \\ T'(k) &= T(\langle 2 \langle g^k \rangle_N \rangle) \\ T''(k) &= T(N-2 \langle g^k \rangle_N) \end{aligned} \quad \text{for } i, k \in \text{INDEX} \quad (12)$$

We note that equations (10) and (11) are in circular correlation form which can be computed easily as there exists a number of fast and easy-implemented algorithms for the realisation of such a structure. Moreover, we can make a further simplification on these equations.

Consider the correlation  $T'(k)$  in eqn.(10) for  $k=1$  to  $N-1$ , we have

$$\begin{bmatrix} T'(1) \\ T'(2) \\ \vdots \\ T'(N-1) \end{bmatrix} = \begin{bmatrix} C(2) C(3) & \cdots & C(1) \\ C(3) C(4) & \cdots & C(2) \\ \vdots & \ddots & \vdots \\ C(1) & \cdots & C(N-1) \end{bmatrix} \begin{bmatrix} x'(1) \\ x'(2) \\ \vdots \\ x'(N-1) \end{bmatrix} \quad (13)$$

$$\text{where } C(n) = \cos [(2\pi/N) \langle g^n \rangle_N] \quad (14)$$

$$\text{As } T'((N-1)/2 + n) = T'(n) \quad \text{for } n=1, 2, \dots, (N-1)/2 \quad (15)$$

only  $T'(0), \dots, T'((N-1)/2)$  are required to compute. One may observe that  $C((N-1)/2 + n) = C(n)$  for  $n=1, 2, \dots, (N-1)/2$ . Hence we have

$$\begin{bmatrix} T'(1) \\ T'(2) \\ \vdots \\ T'((N-1)/2) \end{bmatrix} = \begin{bmatrix} C(2) C(3) & \cdots & C(1) \\ C(3) C(4) & \cdots & C(2) \\ \vdots & \ddots & \vdots \\ C(1) & \cdots & C((N-1)/2) \end{bmatrix} \begin{bmatrix} x'(1) + x'((N+1)/2) \\ x'(2) + x'((N+3)/2) \\ \vdots \\ x'((N-1)/2) + x'(N-1) \end{bmatrix} \quad (16)$$

In other words,  $T'(k)$  is given by:

$$\begin{aligned} T'((N-1)/2 + k) &= T'(k) \\ &= \sum_{i=1}^{(N-1)/2} \{ x'(i) + x'((N-1)/2 + i) \} C(i+k) \end{aligned} \quad \text{for } k=1, 2, \dots, (N-1)/2 \quad (17)$$

This becomes an  $(N-1)/2$  length cyclic correlation.

Similarly,  $T''(k)$  can be realised through another  $(N-1)/2$  length cyclic correlation:

$$\begin{aligned} T''((N-1)/2 + k) &= T''(k) \\ &= \sum_{i=1}^{(N-1)/2} \{ e'(i) + e'((N-1)/2 + i) \} C(i+k) \end{aligned} \quad \text{for } k=1, 2, \dots, (N-1)/2 \quad (18)$$

In summary, we can realise a prime length  $N$  DCT by computing two  $(N-1)/2$  correlations with a cost of  $N-1$  multiplications and  $7(N-1)/2$  additions. This structure is particularly attractive for hardware realisations such as using the distributed arithmetic[6] or other VLSI design techniques.

### 3. AN EXAMPLE

We now clarify our proposal with a length 7 DCT with input sequence  $\{y(i); i=0, 1, \dots, 6\}$ . Obviously, we have

$$Y(0) = y(0) + y(1) + y(2) + y(3) + y(4) + y(5) + y(6)$$

To obtain other  $Y(k)$ 's, we firstly compute the sequence  $x(i)$  from  $y(i)$  with eqn.(2):

$$\begin{aligned} x(6) &= y(6) \\ x(5) &= y(5) - y(6) \\ x(4) &= y(4) - y(5) + y(6) \\ x(3) &= y(3) - y(4) + y(5) - y(6) \\ x(2) &= y(2) - y(3) + y(4) - y(5) + y(6) \\ x(1) &= y(1) - y(2) + y(3) - y(4) + y(5) - y(6) \\ x(0) &= y(0) - y(1) + y(2) - y(3) + y(4) - y(5) + y(6) \end{aligned}$$

Then from eqn.(3), we have

$$Y(1) = [2 T(1) + x(0)] \cos(\pi/14)$$

$$\begin{aligned}
Y(2) &= [2T(2) + x(0)] \cos(2\pi/14) \\
Y(3) &= [2T(3) + x(0)] \cos(3\pi/14) \\
Y(4) &= [2T(4) + x(0)] \cos(4\pi/14) \\
Y(5) &= [2T(5) + x(0)] \cos(5\pi/14) \\
Y(6) &= [2T(6) + x(0)] \cos(6\pi/14)
\end{aligned}$$

By choosing 3 as the primitive root, eqn (17) gives us the results of  $\{T(k); k=2,4,6\}$ :

$$\begin{bmatrix} T^*(4) \\ T^*(5) \\ T^*(6) \end{bmatrix} = \begin{bmatrix} T^*(1) \\ T^*(2) \\ T^*(3) \end{bmatrix} = \begin{bmatrix} T(6) \\ T(4) \\ T(2) \end{bmatrix} = \begin{bmatrix} \cos 4a & \cos 12a & \cos 8a \\ \cos 12a & \cos 8a & \cos 4a \\ \cos 8a & \cos 4a & \cos 12a \end{bmatrix} \begin{bmatrix} x(3) + x(4) \\ x(2) + x(5) \\ x(6) + x(1) \end{bmatrix}$$

for  $a = \pi/7$

$\{T(k); k=1,3,5\}$  can also be obtained from eqn (18)

$$\begin{bmatrix} T^*(4) \\ T^*(5) \\ T^*(6) \end{bmatrix} = \begin{bmatrix} T^*(1) \\ T^*(2) \\ T^*(3) \end{bmatrix} = \begin{bmatrix} T(1) \\ T(3) \\ T(5) \end{bmatrix} = \begin{bmatrix} \cos 4a & \cos 12a & \cos 8a \\ \cos 12a & \cos 8a & \cos 4a \\ \cos 8a & \cos 4a & \cos 12a \end{bmatrix} \begin{bmatrix} x(4) - x(3) \\ x(2) - x(5) \\ x(6) - x(1) \end{bmatrix}$$

Hence two length-3 correlations are required for the computation of a length-7 DCT with a cost of 21 additions and 6 multiplications. These results show that, by comparing with our previously reported results[11], a further reduction in terms of the numbers of multiplications and additions required has been achieved.

#### 4. HARDWARE REALISATION

The proposed algorithm is very structural such that it is very suitable for a VLSI implementation. Particularly, one can realise it with the gate array technique. Figure 1 shows a block diagram of the implementation of the proposed algorithm. To speed up the whole process, pipelining is applied to realise the parallelism. As shown in figure 1, the whole process is divided into 3 stages. In stage 1, the sequence  $\{x(i)\}$  is generated recursively from the input sequence  $\{y(i)\}$ . The value of  $x(0)$  is stored in a register for future usage. The coefficient  $Y(0)$  can be obtained at the same time by summing up the sequence  $\{y(i)\}$ . In this stage, only some simple hardware modules such as adders, registers etc. are required.

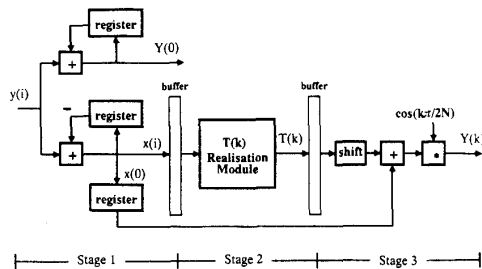


Figure 1. Hardware implementation of the proposed algorithm

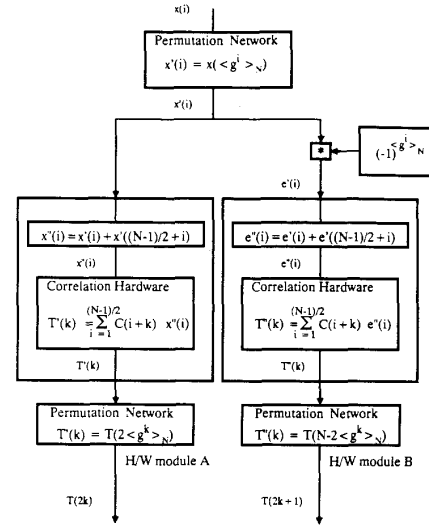


Figure 2. Hardware implementation of  $\{T(k)\}$  sequence.

The hardware module for computing the sequence  $\{T(k)\}$  is shown in figure 2. This module contains permutation modules and two identical hardware modules used for the realisation of the correlation. If one wants to save hardware cost, either one of these two identical modules could be eliminated. However, the realisation time would be unavoidably increased as the workload of the left module would then be double.

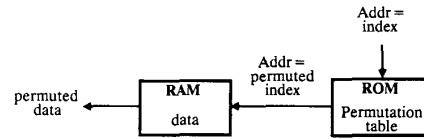


Figure 3. Permutation Network by using Table Look-Up.

The respective permutation networks are realised by using the table lookup technique as shown in figure 3. Comparing with a switch network, the table lookup technique reduces the hardware complexity and hence lowers the hardware cost. In our approach, we use a ROM to store up the address generation table such that we can fetch the permuted data within two memory accesses. It is interesting to point out that in many applications such as image coding, only short length DCT is concerned. Hence, this approach is very efficient and effective as only a small-size table is needed.

The distributed arithmetic technique is used to realise the correlation hardware module. It is recognised that the distributed arithmetic architecture is very suitable for VLSI implementation as it has the following advantages: (1) complicated multiplications can be replaced by a table lookup and shift-add operations, (2) higher accuracy can be achieved by suffering less rounding/truncation errors and (3) modular circuit design is possible. These features allow a high speed circuit design composed of memories, adders and registers only. Figure 4 shows the implementation of the correlation with the distributed arithmetic.

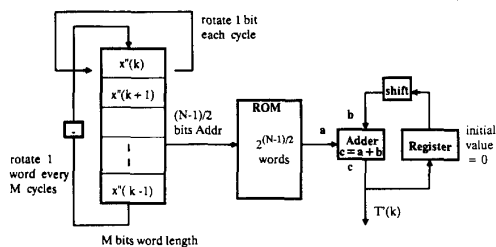


Figure 4. Implementation of convolution with Distributed Arithmetic

After post-permutation, the sequence  $\{T(k)\}$  is fed into the final stage of the implementation. In this stage, items of the sequence will be shifted up by 1 bit and then added to the pre-calculated  $x(0)$  value. The factor  $\cos[k\pi/2N]$  is then multiplied to the corresponding output items to obtain the final cosine transform result. These multiplications can be realised with a multiplier. However, to save the hardware cost further, one can realise it with a shift-add circuit instead. Note that stage 2 dominates the timing of the pipeline process. In stage 2, totally  $M-1$  additions are required for the computation of each  $T(k)$  from the distributed arithmetic processing unit. It means that the time allowed for each multiplication of  $\cos(k\pi/2N)$  term to the corresponding  $T(k)$  is roughly equivalent to the time for  $M-1$  additions. Hence, the most economical and convenient way to realise the multiplication of  $\cos(k\pi/2N)$  would just be a shift-add circuit realised by hardware as shown in figure 5. As the values of  $\cos(k\pi/2N)$ 's are pre-defined, one can construct a table of  $N-1$  words of these values to ease the determination of  $\cos(k\pi/2N)$ . In such case, the processing time of the whole process can be evenly distributed among three stages of the pipe-lined structure.

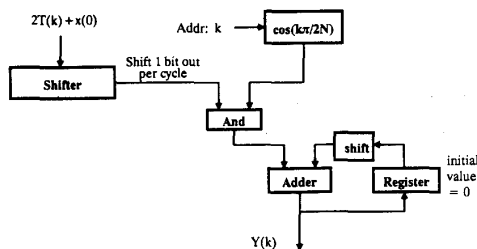


Figure 5. Implementation of the multiplication with shift-add circuit.

These show that the proposed algorithm can be realised efficiently and easily by dedicated hardware or gate array technology. The structure of the hardware required is so simple that it involves only memory and adders. This can achieve a high performance DCT chip at a minimum cost and development time.

## 5. CONCLUSIONS

In this paper, we present a new algorithm such that we can convert a prime length DCT into cyclic correlations. In particular, this algorithm allows us to realise an  $N$ -point DCT by using two  $(N-1)/2$ -point cyclic correlations, where  $N$  is an

odd prime. This makes the realisation simple as the correlation structure is very suitable for hardware realisation using the distributed arithmetic. Finally, by making use of the distributed arithmetic and other simple techniques, we propose an efficient approach to construct a short prime length DCT chip that is suitable for VLSI realisation.

## REFERENCES

- [1] N.Ahmed, T.Natarajan and K.R.Rao, "Discrete cosine transform," *IEEE trans.*, Vol.C-23, pp.90-94, Jan. 1974.
- [2] B.G.Lee, "A new algorithm to compute the discrete cosine transform," *IEEE Trans.*, Vol.ASSP-32, pp.1243-1245, Dec. 1984.
- [3] M.Vetterli and H.Nussbaumer, "A simple FFT and DCT algorithms with reduced number of operation," *Signal Processing*, Vol. 6, No. 4, pp.267-278, August 1984.
- [4] H.S.Hou, "A fast recursive algorithm for computing the discrete cosine transform," *IEEE Trans.*, Vol.ASSP-35, pp. 1455-1461, Oct. 1987.
- [5] S.A.White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," *IEEE ASSP magazine*, Vol.6, No.3, pp.4-19, July 1989.
- [6] T.C.Chen, M.T.Sun and A.M.Gottlieb, "VLSI Implementation of a  $16 \times 16$  DCT," *Proc IEEE 1988*, V1.4, pp.1973-1976.
- [7] W.C.Siu, "Microprocessor-base implementation of digital signal processing using distributed arithmetic," *IERE workshop on Advanced Microprocessors and Digital Signal Processing*, H.K. Section, Sep 1982, pp.148-157.
- [8] W.C.Siu and C.F.Chen, "New realisation technique of high-speed discrete Fourier Transform described by distributed arithmetic," *IEE Proc.*, Vol.130, Pt.E, No.6, Nov 1983, pp.177-182.
- [9] S.Chiu and C.S.Burrus, "A Prime Factor FFT Algorithm using Distributed Arithmetic," *IEEE Trans.*, Vol.ASSP-30, No.2, Apr 1982, pp.217-227.
- [10] Y.H.Chan and W.C.Siu, "A New Convolution Structure for the realisation of Discrete Cosine Transform," *Proceedings, IEEE International Conference on Circuits and Systems (ISCAS'90)*, May 1-3, 1990, New Orleans, U.S.A. pp.2373-2376.
- [11] Y.H.Chan and W.C.Siu, "Algorithm for Prime Length Discrete Cosine Transforms," *Electronics Letters*, 1st Feb 1990, Vol.26, No.3, pp.206-208.