8—4

# Implementing the Discrete Cosine Transform by Using CORDIC Techniques

*Wei–Jou Duh and Ja–Ling Wu*
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, 10764, Taiwan, R.O.C.

**Abstract**

In this paper, a new Constant–Rotation DCT processor architecture using CORDIC techniques, CRDCT, is proposed. This newly proposed DCT architecture is composed of a linear array of identical CORDIC processing elements. Since each processing element is identical, this architecture is especially suitable for VLSI implementation and the costs of design and implementation are reduced. The breadboard implementation of 8–point Constant–Rotation DCT is also given in this paper to verify the correctness of the CRDCT.

## I. Introduction

In the decade since the introduction of discrete cosine transform (DCT) [1], it has found a number of applications in image processing [2] and, more recently in speech processing [3]. It has been shown that the DCT performs very close to the statistically optimal Karhunen–Loeve transform (KLT) for a large number of signal classes [4]–[5]. Therefore, many special architectures for computing DCT have been invented. Since it is used mostly in various signal processing applications, the DCT processors with real–time computation capabilities are required urgently. Some DCT architectures [6,7] were proposed to meet this requirement.

In the recent paper, a novel concurrent CORDIC architecture [8] for computing DCT has been proposed by the authors. Although the concurrent DCT architecture is very attractive due to its high throughput rate; however, it requires N/2 different CORDIC processors. From the view points of design and implementation, such an approach is not cost–effective. In order to overcome this drawback, a new constant–rotation DCT (CRDCT) architecture, based on CORDIC techniques, is presented in this paper. Figure 1 shows the system block diagram of the architecture, and Fig.2 shows the inner structure of the so–called 'CRDCT processor'. From Fig.2, it is clear that CRDCT processor is composed of a linear array of identical PEs (constant–rotation CORDIC processor) combined with some 'accumulator–rings'. Because all PEs are identical, this design is especially suitable for pratical VLSI implementation of high speed DCT chips.

This paper is organized as follows: Section II describes the preliminary mathematical backgrounds ,called 'index partitions' [8]. Section III presents the CRDCT architecture. The results of breadboard implementation of an 8–point CRDCT processor is given in section IV. Conclusions and discussions are presented in the final section. Proofs of selected lemmas are given in Appendix.

## II. The Index Partitions

The DCT of a real input sequence x(k) is defined by

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

where N is the transform length. The corresponding inverse DCT (IDCT) is defined as

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

where c(n) is given by

$$\begin{aligned} c(n) &= \frac{1}{\sqrt{2}} \quad ,\text{for } n=0 \\ &= 1 \quad ,\text{otherwise} \end{aligned} \quad (3)$$

In order to achieve high speed computation of DCT, a specific index partition is performed. In what follows, we assume that the transform length N equals to $2^m$, where m is an integer, and denote n and k, respectively, as the indices of output and input. The output index n is now partitioned into m subsets as follows. Let $Z_N$ be the set {0,1,...,N–1}. The following lemma forms the basis of the index partition given in corollary 1.

**Lemma 1:** The integer set $Z_N$ is the disjoint union of the integer subsets,

$$A_i = \{ 2i(2j+1) \}$$

where j is in $\{0,1,\ldots,2^{m-i-1}-1\}$ and i in {0,...,m–2}, and

$$A_{m-1} = \{ 0,2^{m-1} \} \quad (4)$$

then

$$Z_N = \bigcup_{i=0}^{m-1} A_i$$

where $A_i \cap A_j$ is an empty set, for $i \neq j$.

For the ease of explanation, some functions and sets are defined in the following. Let $W_N$ and $R_N$ denote the sets { 1,3,5,...,2N–1 } and { 0,1,2,...,N/2–1 }, respectively. Let g be a function defined from $Z_N$ to $W_N$ by

$$g(x) = 2x + 1 \quad (5.a)$$

where x belongs to $Z_N$, and $f_N$ be a function from $W_N x Z_N$ to $R_N$ by

$$f_N(k,n) = N - (kn \bmod N), \text{if } kn > N/2 \quad (5.b)$$
$$= kn \bmod N \quad ,\text{otherwise}$$

where k belongs to $W_N$ and n belongs to $Z_N$.

Corollary 1: The integer set $R_N$ is the disjoint union of the integer subsets,

$$B_i = \{ 2i(2j+1) \}$$

where j is in $\{ 0,1,...,2^{m-i-2}-1 \}$ and i in $\{ 0,...,m-3 \}$

$$B_{m-2} = \{ 2^{m-2} \}$$
$$B_{m-1} = \{ 0 \} \quad (6)$$

and

$$R_N = \bigcup_{i=0}^{m-1} B_i ,$$

where $B_i \cap B_j$ is an empty set, for $i \neq j$.

Since CORDIC processors [9,10] can be used to compute sine and cosine functions simultaneously, the transform kernels of DCT can be changed to pairs of these two functions. The function $f_N$ defined in (5.b) is used to map the input indices, say k, and output indices, n, to the corresponding PE. Obviously, $W_N$ is the set of input indices, $Z_N$ is the output indices set, and $R_N$ is the set of PE's identification address.

On the basis of the symmetric property of DCT given in [8, eqn.(14)], Y(n) and Y(N–n) is a sine–cosine function pair. In fact, each sine–cosine function pair corresponds to an accumulator pair. Hence, there are N/2 function pairs and N/2 CORDIC processors are required. In the following section, we will modify the concurrent CORDIC DCT processor [8] into a more cost–effective one, the so–called CRDCT processor.

## III. The Constant–Rotation CORDIC DCT Architecture

According to the coordinate rotation concepts [9,10], rotation by $i\theta$ equals to rotation by $\theta$ i times successively. This leads to the CRDCT architecture which cascades identical CORDIC processors in the form of linear array.

### A. System Overviews

Since Y(n) and Y(N–n) is a function pair [8], only the first half of the output indices need to be considered. Furthermore, the first half of the output index set, say $Z_{N/2}$, equals to $R_N$. From Corollary 1, the output index set $Z_{N/2}$ can be decomposed into m subsets and one can obtain the following lemma intuitively.

**Lemma 2:** For any x in $W_N$ and y in $B_i$, and

$$f_N(x,y) = r$$

then r is also in $B_i$.

From Lemma 2, each element in subset $B_i$ can be mapped to the same set $B_i$ via the function $f_N$. Since the set $Z_{N/2}$ denotes the output index set and $R_N$ denotes the address of PE, the implication of Lemma 2 is that for any given input with index x, the results of the y–th output accumulator pair, where y belongs to $B_i$, are computed by those PEs with address r in the same $B_i$. Therefore, there are m accumulator–rings in this architecture.

In order to meet the timing requirements of the accumulator–rings, there is a side effect that both inputs and outputs need permutations. The input and output permutations are described in the following subsections, respectively.

### B. Input Reordering

From the symmetric properties of cosine and sine functions, for each n, the inputs x(k), x(N/2–k–1), x(N/2+k), and x(N–k–1) should be computed at the same PE. Therefore, only the first quarter of input indices will be considered. The rest part of input can be reordered according to the symmetric properties. Hence, the input index set is reduced to $W_{N/4}$.

The longest accumulator–ring is used to determine the input sequence of CRDCT. Surprisingly, the first quarter of input index set, say $W_{N/4}$, and the largest subset, $B_0$ of $R_N$, are equal. Now a binary operator $\circledast_N$ is defined over $B_0$ to form a cyclic group as follows:

Let $G_N = \{ B_0 ; \circledast_N \}$ be a group, where $\circledast_N$ is a binary operator defined over $B_0$ and the definition of $\circledast_N$ is given by

$$x \circledast_N y = N - (xy \bmod N), \text{ if } xy > N/2 \quad (8)$$
$$= xy \quad , \text{otherwise}$$

Table 1 shows the operations of $\circledast_{16}$. Since we want to simplify the data transfers between these accumulators, the data stored in these accumulators are shifted along the ring–paths in each clock period. Thus for a given PE, the operation table of group $G_N$ must in the form as Table 2. Note that Table 2 exhibits itself as a cyclic group [11]. Therefore, if one can find, at least, one generator in group $G_N$, then the generator can generate the required input sequence. Table 3 shows the case of N=16. The following lemma shows that 3 is a generator of $G_N$ and thus complete the input ordering processes.

**Lemma 3:** 3 is a generator of group $G_N$.

Proof: See Appendix–1.

Since the above consideration concerns only on the first quarter of the input indices, the overall input sequence is constructed using the symmetric property. The last step in designing this architecture is to add some delay buffers in appropriate PEs to ensure the correctness of the timing. An example of 16–point CRDCT is given in Fig.3.

### C. Output Reordering

Once the input sequence order is determined, the output sequence is also determined. Because the output sequence of each ring can be found from Table 2, it is obviously that output sequence is also generated by a generator. Thus, 3 is used to generate the output sequence, too. The function $h_i$ is first defined as follows.

$$h_i(x) = 2^i x \tag{9}$$

The output sequences of accumulator–rings associated with the index set $B_i$ is $h_i(3^j$ under $\otimes_N)$, for $j=0,1,\ldots,2^{m-2-i}-1$.

For example, the 16–point CRDCT processor consists of 4 sets of accumulators. The output order corresponds to $B_0$ is $h_0(3^j$ under $\otimes_{16})$, for j=0,1,2,3, i.e. { 1,3,7,5 }, and so forth. Thus, refer to Fig.3, these accumulators can be assembled according to their locations. Because the accumulators appear in pairs, the order of the final results is { 1,15,2,14,3,13,4,12,7,9,6,10,5,11,0,8 }.

## IV. The Breadboard Implementation

Since the newly proposed CRDCT architecture is suitable for implementation, an 8–point CRDCT is implemented in breadboards. The block diagram of this implementation is shown in Fig.4. The input device is a 4x4 keypad with hexadecimal number. The output devices are 7–segment LED displays.

This implementation takes 8–bits input data and produces 8–bits output DCT coefficients. The internal precision for computation is 12–bits. The breadboard implementation proves that the round off error is within 1/256. Each CORDIC processor is identical and with rotation angle $\pi/16$, the picture of a CORDIC PE is given in Fig.5. The block diagram of the designed CORDIC processor is shown in Fig.6.

The approximating values of Cos[$\pi$/16] and Sin[$\pi$/16] are 251/256 and 50/256, respectively. Only MSI and SSI are used in this implementation, such as inverters, adders, multiplexers, and registers. Each CORDIC processor is constructed by 64 TTL chips. The overall breadboard implementation is shown in Fig.7. The theoretical maximum clock rate is about 12MHz and the maximum working clock rate is 8MHz.

## V. Conclusions and Discussions

In this paper, a new CORDIC–based DCT processor, CRDCT, is proposed. This new DCT processor is composed of a linear array of identical CORDIC processors combined with some 'accumulator–rings'. Because each PE is identical, the cost of design and implementation are reduced. The breadboard implementation verifies the correctness of the proposed approach.

## Appendix–1

Proof of Lemma 3:

We will prove the lemma by proving $3^{\phi(N/2)-1} = -1$ mod N/2, where $\phi(.)$ is Euler's $\phi$ function. Since $\otimes$N is similar to the operation mod N/2 and $(N/2-1)^2 = 1$ (mod N/2), then the proof will be $3^{N/8} = N/2 - 1$. Note that $3^{N/8} = (1 + 2)^{N/8}$ and the expansion of the formula is:

$$\begin{aligned}3^{N/8} &= (1+2)^{N/8}\\ &= 1 + N/4 + (N/4)(N/8-1) + (N/2)(N/8-1)(N/8-2)/3 +\\ &\quad (N/4)(N/8-1)(N/8-2)(N/8-3)/3 + \ldots + 2^{N/8}\end{aligned}$$

Consider the expansion under the operation $\otimes_N$, we first take modulo N. The expansion becomes

$$\begin{aligned}&1 + 2^{2m-5} - 2^{m-1} \text{ (mod N)}\\ &= 1 - 2^{m-1} \text{ (mod N)}\\ &= 2^{m-1} + 1 \text{ (mod N)} \quad > N/2\end{aligned}$$

Thus,

$$\begin{aligned}3^{N/8} &= N - (2^{m-1} + 1)\\ &= N/2 - 1 \doteq -1,\end{aligned}$$

and the proof is complete.

| $\otimes_{16}$ | 1 | 3 | 5 | 7 |
|---|---|---|---|---|
| 1 | 1 | 3 | 5 | 7 |
| 3 | 3 | 7 | 1 | 5 |
| 5 | 5 | 1 | 7 | 3 |
| 7 | 7 | 5 | 3 | 1 |

Table 1.

| $\otimes_{16}$ | e | a | b | c |
|---|---|---|---|---|
| e | e | a | b | c |
| a | a | b | c | e |
| b | b | c | e | a |
| c | c | e | a | b |

where b is $a^2$ and c is $a^3$

Table 2.

| $\otimes_{16}$ | 1 | 3 | 7 | 5 |
|---|---|---|---|---|
| 1 | 1 | 3 | 7 | 5 |
| 3 | 3 | 7 | 5 | 1 |
| 7 | 7 | 5 | 1 | 3 |
| 5 | 5 | 1 | 3 | 7 |

Table 3.

## Reference

[1] N.Ahmed, T.Natarajan, and K.R.Rao, "Discrete Cosine Transform", IEEE Trans. on Comput., Vol.C–25, pp.90–93, Jan. 1974.

[2] A.K.Jain, "Image Data Compression: A Review", Proc. IEEE, pp.349–389, 1981.

[3] R.Zelinski and P.Noll, "Adaptive Transform Coding of Speech Signals", IEEE Trans. on Acoustic, Speech, Signal Processing, Vol.ASSP–25, pp.299–309, Ayg. 1977.

[4] M.Hamidi and J.Pearl, "Comparison of the Cosine and Fourier Transforms of Markov–1 Signals", IEEE Trans. on Acoustic, Speech, Signal Processing, Vol.ASSP–24, pp.428–429, Oct. 1976.

[5] R.J.Clark, "Relation between the Karhunen–Loeve and Cosine Transform", IEE Proc., Vol.128, Part–F, pp.359–360, Nov. 1981.

[6] M.T.Sun, L.Wu, and M.L.Liou, "A Concurrent Architecture for VLSI Implementation of Discrete Cosine Transform" IEEE Trans. Circuits and Systems, Vol.CAS–34, No.8, pp.992–994, Aug. 1987.

[7] M. Vetterli and Adriaan Ligtenberg, "A Discrete Fourier–Cosine Transform Chip" IEEE Journal on Selected Areas in Communications, Vol.SAC–4, No.1, pp.49–61, Jan. 1986.

[8] Ja–Ling Wu and Wei–Jou Duh, "A Novel Concurrent architecture to Implement Discrete Cosine Transform Based on Index Partitions", to be appeared in International Journal of Electronics.

[9] J. E. Volder, "The CORDIC Trigonometric Computing Technique" IRE Trans. Electronic Computers, Vol.EC–8, No.3, pp.330–334, Sept.1959.

[10] Erling H. Wold and Alvin M. Despain, "Pipeline and Parallel–Pipeline FFT Processors for VLSI Implementations", IEEE Trans. Comput., Vol.C–33, No.5, pp.414–426, May 1984.

[11] "Elements of Modern Algebra" by J.Gilbert and L.Gilbert, PWS Publishers, 1984.

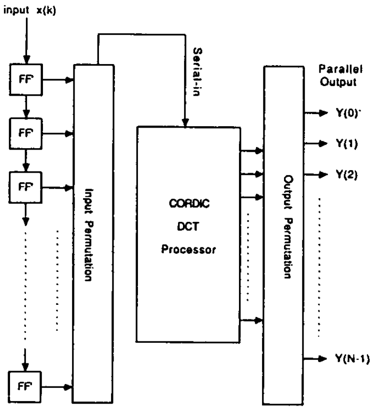

Fig. 1 Block Diagram of CORDIC DCT Architecture

* FF means data latch

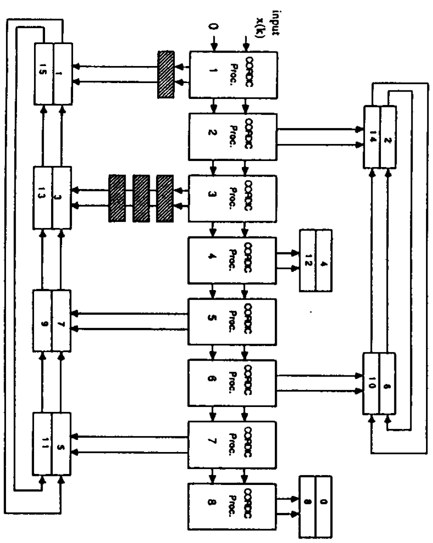

Fig. 3 A 16-point Constant-Rotation DCT Architecture

▨ Delay Buffer

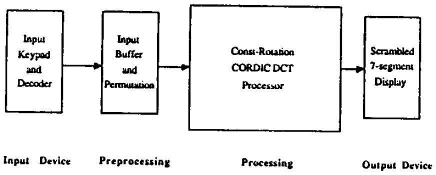

Figure 4. The block-diagram of the CRDCT Breadboard Implementation

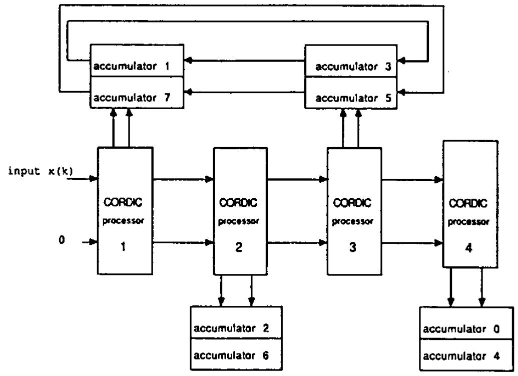

Figure 2. An 8-point Constant-Rotation DCT Architecture

Figure 5. The CORDIC Processing Element

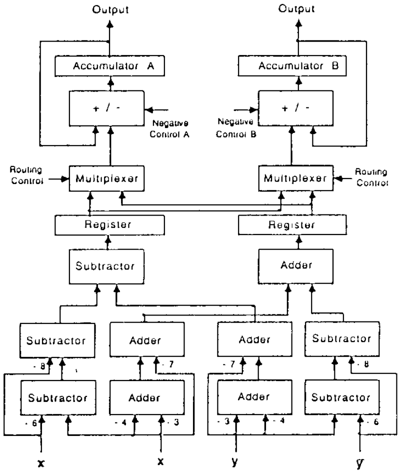

Figure 6 A specific CORDIC processor with rotation angle $\frac{\pi}{16}$ is given here, where '-3' denotes shift left 3 bits.

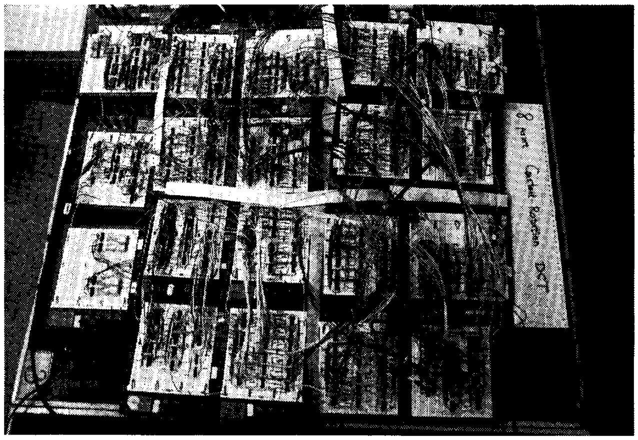

Figure 7. The overall breadboard implementation.