International Journal of Electronics
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713599654

Constant-rotation DCT architecture based on CORDIC techniques

Wei-Jou Duh [a]; Ja-Ling Wu [a]
[a] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

Online Publication Date: 01 November 1990

To cite this Article Duh, Wei-Jou and Wu, Ja-Ling(1990)'Constant-rotation DCT architecture based on CORDIC techniques',International Journal of Electronics,69:5,583 — 593
To link to this Article: DOI: 10.1080/00207219008920343
URL: http://dx.doi.org/10.1080/00207219008920343

INT. J. ELECTRONICS, 1990, VOL. 69, NO. 5, 583–593

Constant-rotation DCT architecture based on CORDIC techniques

WEI-JOU DUH† and JA-LING WU†

A new constant-rotation DCT (CRDCT) processor architecture using CORDIC techniques is proposed. This processor is composed of a linear array of identical CORDIC processors combined with some 'accumulator-rings'. Since each processing element is identical with every other, this architecture is especially suitable for VLSI implementation, and the cost of design and implementation is reduced. The breadboard implementation of an eight-point constant-rotation DCT processor is also described for verification of the performance of the CRDCT.

1. Introduction

In the 15 years since the introduction of the discrete cosine transform (DCT) (Ahmed *et al.* 1974) it has found a number of applications in image processing (Jain 1981) and speech processing (Zelinski and Noll 1977). It has been shown that the DCT's performance is very close to that of the statistically optimal Karhunen–Loéve transform (KLT) for a large number of signal classes (Hamidi and Pearl 1976, Clark 1981). Many fast algorithms for computing DCT have been published. A comprehensive review of various DCT algorithms is given by Makhoul (1980) and a novel recursive DCT algorithm is presented by Hou (1987). Since they are used mostly in various signal processing applications, DCT processors with inbuilt real-time computation capabilities are urgently needed. Some DCT architectures were proposed to meet this requirement, such as the distributed arithmetic DCT by Sun *et al.* (1987) and the discrete Fourier–cosine transform chip by Vetterli and Ligtenberg (1986).

In a recent paper (Wu and Duh 1990) the present authors proposed a concurrent CORDIC architecture for computing DCT. Although the concurrent DCT architecture is very attractive for its high throughput rate, it requires $N/2$ CORDIC processors with different rotating angles. This implies that each CORDIC processor is different from every other. From the viewpoint of design and implementation, the approach is not cost effective. To overcome this drawback a new constant-rotation DCT (CRDCT) architecture, based on CORDIC processors, is presented in this paper. Basically, it is a serial-in parallel-out architecture. Figure 1 shows the block diagram of the proposed DCT architecture, and Fig. 2 the inner structure of the so-called 'CRDCT processor'. From Fig. 2, it is clear that the CRDCT processor is composed of a linear array of identical CORDIC processors (constant-rotation CORDIC processors) combined with some 'accumulator-rings'. Because every processing element (CORDIC processor) is identical this design is especially suitable for VLSI implementation of high-speed DCT chips.

This paper is organized as follows. Section 2 describes the preliminary mathematical background behind the proposed architecture—the so-called 'index partitions' (Wu and Duh 1990). Section 3 describes the CRDCT architecture, which is

Received 7 April 1989; accepted 12 July 1989.

† Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 10764, Taiwan, Republic of China.

0020-7217/90 $3.00

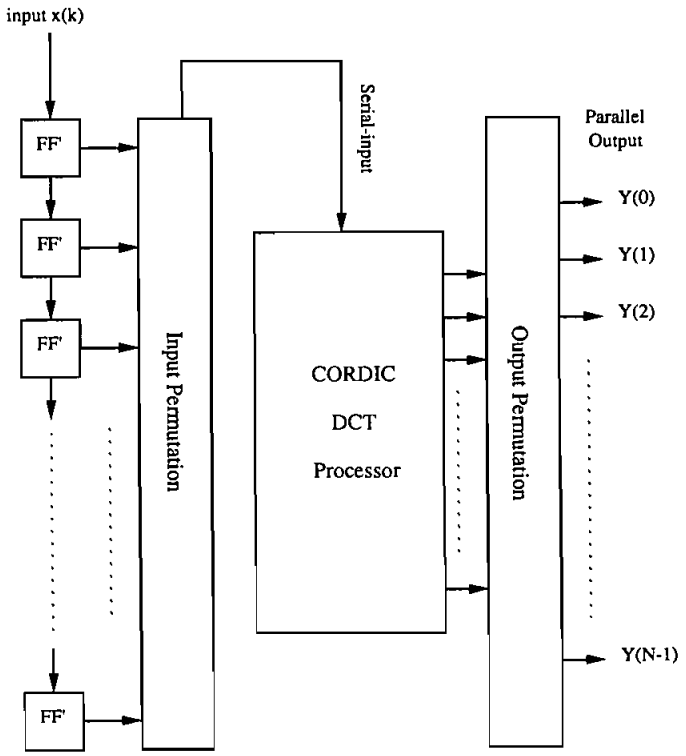

Figure 1. Block diagram of CORDIC DCT architecture (FF—data latch).

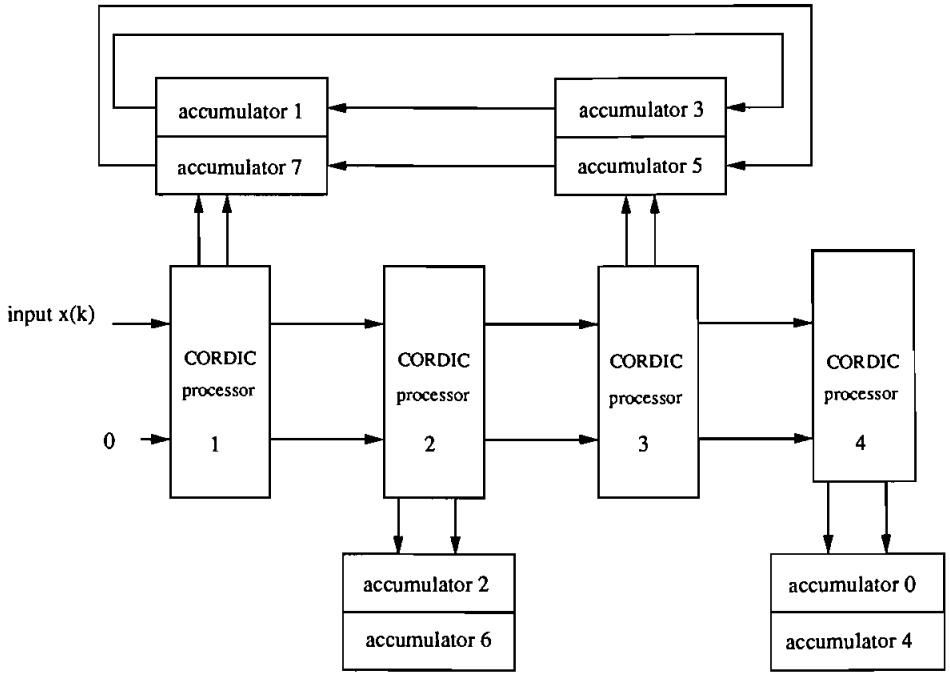

Figure 2. Eight-point constant-rotation DCT architecture.

simple and regular for VLSI implementations. The results of breadboard implementation of an eight-point CRDCT processor are given in §4. Conclusions and a discussion are presented in the final section. Proofs of selected lemmas are given in the appendices.

2. Index partitions

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

$$Y(n) = 2c(n)/N \sum_{k=0}^{N-1} x(k) \cos [\pi(2k + 1)n/2N] \tag{1}$$

where N is the transform length, $\pi = 3{\cdot}14159 \ldots$, and $n, k = 0, 1, \ldots, N - 1$. Correspondingly, the definition of the inverse DCT (the IDCT) is

$$x(k) = \sum_{n=0}^{N-1} c(n)Y(n) \cos [\pi(2k + 1)n/2N] \tag{2}$$

where $c(n)$ is given by

$$c(n) = \begin{cases} 1/\sqrt{2}, & \text{for } n = 0 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

To achieve high-speed computation of DCT, a specific index partition is performed. In what follows, we assume the transform length N equals to 2^m, where m is an integer; n and k, respectively, denote output and input indices. The output index n is now partitioned into m subsets as follows: let Z_N be the set $\{0, 1, \ldots, N - 1\}$. The following lemma shows that Z_N can be decomposed into m disjoint subsets, and 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

$$\left.\begin{aligned} A_i &= \{2^i(2j + 1) \mid j \in \{0, 1, \ldots, 2^{m-i-1} - 1\}, \text{ and } i \in \{0, \ldots, m - 2\}\} \\ A_{m-1} &= \{0, 2^{m-1}\} \end{aligned}\right\} \tag{4}$$

Then

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

where the intersection of A_i and A_j is an empty set for $i \neq j$.

Proof

For the proof, see Appendix A.

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

$$g(x) = 2x + 1 \tag{5 a}$$

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

$$f_N(k, n) = \begin{cases} N - (k \cdot n \bmod N), & \text{if } k \cdot n \geqslant N/2 \\ k \cdot n \bmod N & \text{otherwise} \end{cases} \tag{5 b}$$

where k belongs to W_N and n belongs to Z_N. Further, from Lemma 1, the set R_N can also be decomposed into m subsets, as shown in the following corollary.

Corollary 1

The integer set R_N is the disjoint union of the integer subsets

$$\left.\begin{aligned} B_i &= \{2^i(2j+1) \mid j \in \{0, 1, \ldots, 2^{m-i-2}-1\}, \text{ and } i \in \{0, \ldots, m-3\}\} \\ B_{m-2} &= \{2^{m-2}\} \\ B_{m-1} &= \{0\} \end{aligned}\right\} \quad (6)$$

and

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

where the intersection of B_i and B_j is an empty set for $i \neq j$.

Since CORDIC processors (Volder 1959, Wold and Despain 1984) can be used to compute sine and cosine functions simultaneously, the transform kernels of DCT can be changed to pairs of these two functions. Each pair of sine–cosine functions can be computed at the same processing element to correspond to a specific CORDIC processor. The function f_N defined in (5 *b*) is used to map the input indices, say k, and output indices, say n, to the corresponding processing elements. Obviously, W_N is the set of input indices, Z_N is the output indices set, and R_N is the set of PEs' identification address.

On the basis of the symmetric property of DCTs given by Wu and Duh (1990 equation (14)), $Y(n)$ and $Y(N-n)$ are 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 thus $N/2$ CORDIC processors are required. In the following section we will modify the CORDIC DCT processor of Wu and Duh (1990) into a more cost-effective one, which we call a CRDCT processor.

3. Constant-rotation CORDIC DCT architecture

For simplicity we denote cos $[i \cdot \pi/2N]$ by C^i and sin $[i \cdot \pi/2N]$ by S^i. According to the coordinate rotation concepts of Volder (1959) and Wold and Despain (1984), and rotation by C^2 equals twice the rotation by C^1. Similarly, rotation by C^i is equivalent to rotation by C^1 i times successively, and the same for S^i. This leads to the constant-rotation DCT architecture which cascades identical CORDIC processors in the form of a linear array, as shown in Fig. 2.

3.1 *System overviews*

Since $Y(n)$ and $Y(N-n)$ are a function pair (Wu and Duh 1990), i.e. they can be computed at the same PE, only the first half of the output indices need be considered. That is, the first half of the output index set, say $Z_{N/2}$, equals R_N. By using the results of Corollary 1, the output index set $Z_{N/2}$ can be decomposed into m subsets, say $B_0, B_1, \ldots, B_{m-1}$. Also from Corollary 1, one can obtain the following lemma intuitively.

Lemma 2

For any x in W_N and y in B_i, and

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

r is also in B_i.

$\otimes_{16}$	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.

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 identification address of the PE, the implication of Lemma 2 is that there is a particular set of PEs corresponding to a particular set of accumulators. That is, for any given input with index x the results of the yth output accumulator pair, where y belongs to B_i, are computed by those PEs with the identification address r in the same B_i. That is why there are m accumulator-rings in this newly proposed architecture.

To meet the timing requirements of the accumulator-rings there is a side-effect that both inputs and outputs need permutations. Since the CRDCT architecture operates in a serial-in parallel-out mode, the input and output permutations are described in the following subsections, respectively.

3.2. *Input reordering*

Now consider the following equation:

$$\begin{aligned} Y(n) = \sum_{k=0}^{N/4-1} \{ & x(k) \cos [\pi(2k+1)n/2N] \\ & + x(N/2-k-1) \cos [n \cdot \pi/2 - \pi(2k+1)n/2N] \\ & + x(N/2+k) \cos [n \cdot \pi/2 + \pi(2k+1)n/2N] \\ & + x(N-k-1) \cos [n \cdot \pi - \pi(2k+1)n/2N] \} \end{aligned} \tag{7}$$

Equation (7) implies that 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, because they are multiplied by the cosine or sine functions of the same angle. Henceforth only the first quarter of the input indices will be considered. The remainder of the input can be reordered according to the symmetric property illustrated in (7). Therefore the input index set is reduced to $W_{N/4}$.

The longest accumulator-ring is used to determine the input sequence of CRDCT. Since A_0 is the corresponding set of the longest accumulator-ring, only B_0 need be considered. Surprisingly, the first quarter of the input index set, denoted by $W_{N/4}$, and the largest subset, B_0 of R_N, are equal. Now a binary operator $\otimes_N$ is defined over B_0 to form a cyclic group as follows.

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

$$x \otimes_N y = \begin{cases} N - (x \cdot y \bmod N), & \text{if } x \cdot y > N/2 \\ x \cdot y, & \text{otherwise} \end{cases} \tag{8}$$

where x and y belong to B_0. Table 1 shows the operations of $\otimes_{16}$.

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

Note. b is a^2 and c is a^3

Table 2.

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, i.e. a fixed identification address, the operation table of group G_N must take the form shown in Table 2. Note that Table 2 proves to be a cyclic group (Gilbert and Gilbert 1988), with 'a' as the generator and 'e' the identity. Therefore, if one can find at least one generator in group G_N the generator can generate the required input permutations. Table 3 shows the case for $N = 16$. The following lemma states that 3 is a generator of G_N and thus completes the input ordering processes.

Lemma 3

3 is a generator of group G_N.

Proof

For the proof, see Appendix B.

Since the above consideration concerns only the first quarter of the input indices, the overall input sequence order is constructed using the symmetric property given in (7). Table 4 lists the input orders for some different transform lengths. 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 a 16-point CRDCT is given in Fig. 3.

$\otimes_{16}$	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.

Point no.	Input order
8	0, 1, 3, 2, 4, 5, 7, 6
16	0, 1, 3, 2, 7, 6 , 4, 5, 8, 9, 11, 10, 15, 14, 12, 13
32	0, 1, 4, 2, 7, 6, 3, 5, 15, 14, 11, 13, 8, 9, 12, 10, 16, 17, 20, 18, 23, 22, 19, 21, 31, 30, 27, 29, 24, 25, 28, 26
64	0, 1, 4, 13, 8, 6, 12, 5, 15, 14, 11, 2, 7, 9, 3, 10, 31, 30, 27, 18, 23, 25, 19, 26, 16, 17, 20, 29, 24, 22, 28, 21, 32, 33, 36, 45, 40, 38, 44, 37, 47, 46, 43, 34, 39, 41, 35, 42, 63, 62, 59, 50, 55, 57, 51, 58, 48, 49, 52, 61, 56, 54, 60, 53

Table 4.

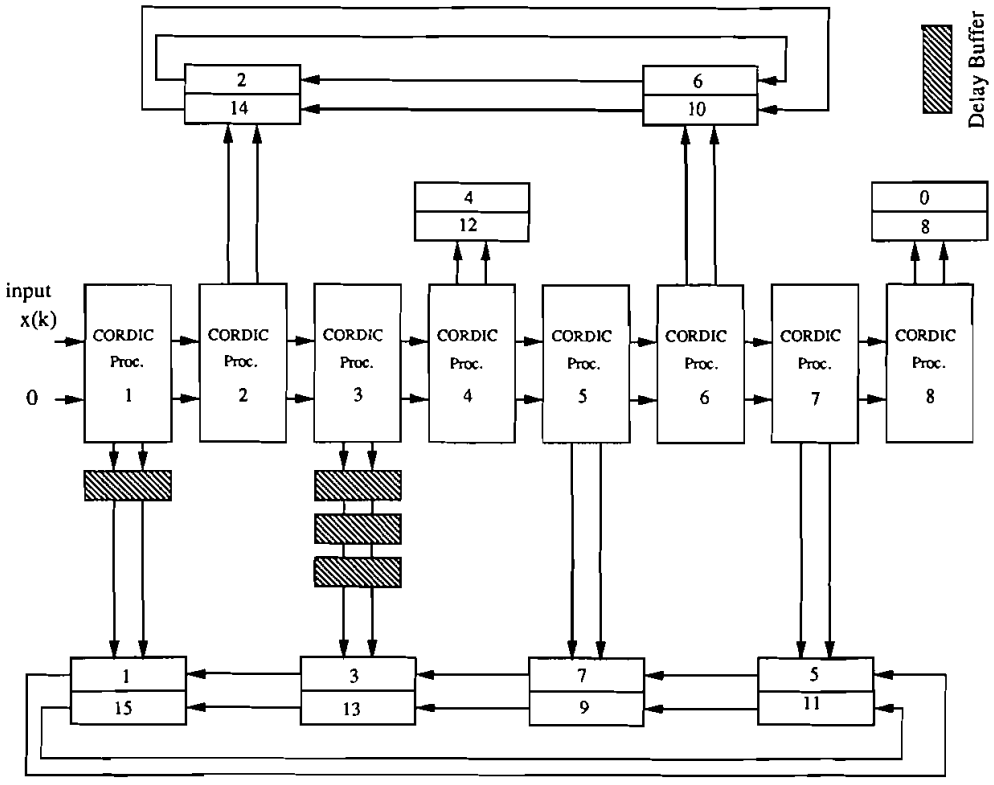

Figure 3. Sixteen-point constant-rotation DCT architecture.

3.3. *Output reordering*

Once the input sequence order is determined the output sequence corresponding to each accumulator-ring is also determined, since the output sequence order of each ring can be found from Table 2 whenever the input sequence is set.

It is obvious that the output sequence is also generated by a generator from Table 3. Thus three is also used to generate the output sequence. To formulate the output order, the function h_i is first defined as follows:

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

The output sequence of the accumulator-ring 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 clarity let us consider the case of $N = 16$ in detail. The 16-point CRDCT processor consists of four sets of accumulators corresponding to B_0, B_1, B_2, and B_3, respectively. The output order corresponding to B_0 is h_0 (3^j under $\otimes_{16}$), for $j = 0, 1, 2, 3$, i.e. $\{1, 3, 7, 5\}$; to B_1 is h_1 (3^j under $\otimes_{16}$), for $j = 0, 1$, i.e. $\{2, 6\}$; to B_2

Point no.	Output order
8	1, 7, 2, 6, 3, 5, 0, 4
16	1, 15, 2, 14, 3, 13, 4, 12, 7, 9, 6, 10, 5, 11, 0, 8
32	1, 31, 2, 30, 3, 29, 4, 28, 9, 23, 6, 26, 5, 27, 8, 24, 15, 17, 14, 18, 13, 19, 12, 20, 7, 25, 10, 22, 11, 21, 0, 16

Table 5.

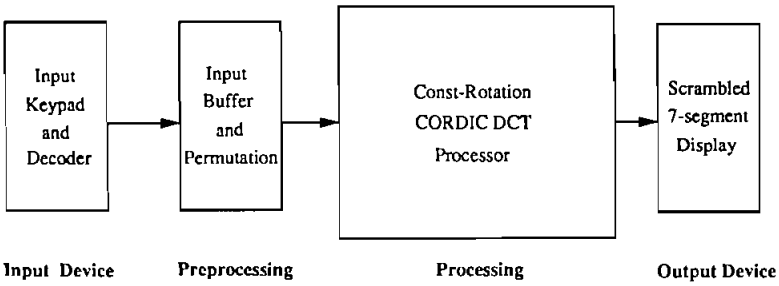

Figure 4. Block-diagram of CRDCT breadboard implementation.

is h_2 (3^0 under $\otimes_{16}$), i.e. $\{4\}$; to B_3 is $\{0\}$. Thus (Fig. 3) these accumulators can be assembled according to their locations. That is, for B_i accumulators the output h_i (3^j under $\otimes_N$) is located at the CORDIC processor numbered $h_i(2j + 1)$. Hence the output order becomes $\{1, 2, 3, 4, 7, 6, 5, 0\}$ associated with the CORDIC processors numbered 1, 2, 3, 4, 5, 6, 7, 8. Because the accumulators appear in pairs the order of the final results is $\{1, 15, 2, 14, 3, 13, 4, 12, 7, 9, 6, 10, 5, 11, 0. 8\}$. Table 5 presents the output orders for some different transform lengths in physical use.

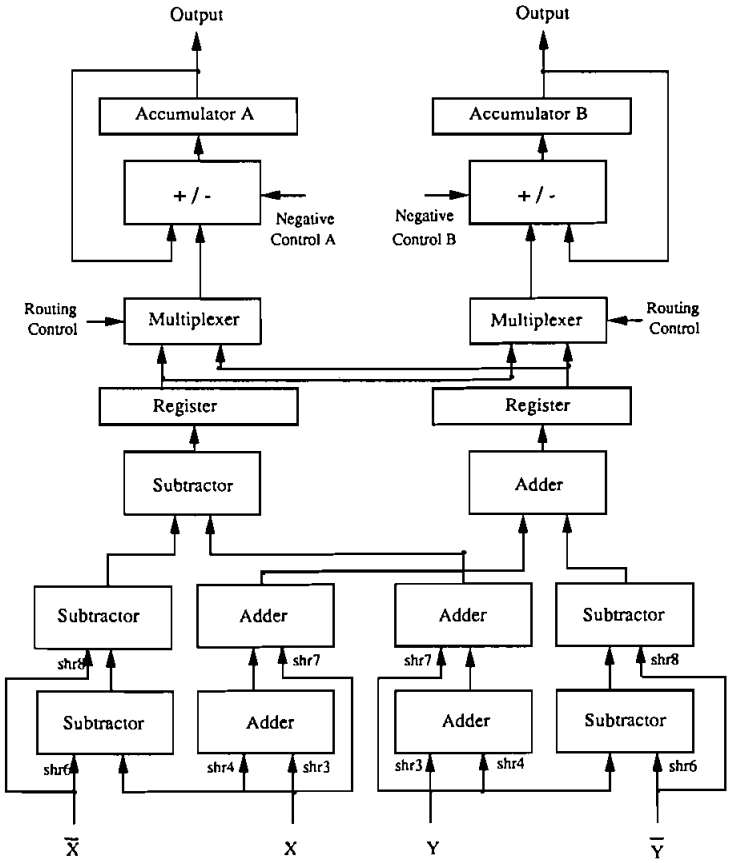

Figure 5. Specific CORDIC processor with rotation angle $\pi/16$ ('shr$p$' denotes 'shift right p bits').

Figure 6. View of overall breadboard implementation.

4. Breadboard implementation

Since the newly proposed CRDCT architecture is suitable for implementation, an eight-point CRDCT is implemented on breadboards. The block diagram of this implementation is shown in Fig. 4. The input device is a 4 × 4 keypad with hexadecimal numbering. The output devices are seven-segment LED displays.

This implementation takes eight-bits input data and produces eight-bits output DCT coefficients. The internal precison for computation is twelve-bits. The breadboard implementation proves that the round-off error is within 1/256. Every CORDIC processor is identical, with a rotation angle $\pi/16$. The block diagram of the CORDIC processor design is shown in Fig. 5.

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 from 64 TTL chips. The overall breadboard implementation is shown in Fig. 6. The theoretical maximum clock rate is about 12 MHz and the maximum working clock rate is 8 MHz.

5. Conclusions and discussion

In this paper a new CORDIC-based DCT processor, CRDCT, has been 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 with every other, the costs of design and implementation are reduced. The breadboard implementation confirms the correctness of the proposed approach.

Appendix A

Proof of Lemma 1

Since $N = 2^m$, each element of Z_N belongs to a subset A_i. Let the subset $A_i = \{x \mid \gcd(N, x) = 2^i\}$, where $i = 0, 1, 2, \ldots, 2^{m-2}$, and $A_{m-1} = \{0\} \cup \{x \mid \gcd(N, x) = 2^{m-1}\}$. Thus Z_N is the union of disjoint subsets A_i, where $i = 0, 1, \ldots, m-1$.

Appendix B

Proof of Lemma 3

We prove the lemma by proving $3^{\phi(N/2)-1} = -1 \bmod N/2$, where $\phi(\cdot)$ is Euler's ϕ function. Since $\otimes_N$ is similar to the operation mod $N/2$ and $(N/2-1)^2 \equiv 1 \pmod{N/2}$, then the proof will be $3^{N/8} = N/2 - 1$. Note that

$$3^{N/8} = (1+2)^{N/8}$$

the expansion of the formula is

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

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

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

Thus,

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

and the proof is complete.

References

Ahmed, N., Natarajan, T., and Rao, K. R., 1974, Discrete Cosine Transform. *I.E.E.E. Transactions on Computers*, **25**, 90–93.

Clark, R. J., 1981, Relation between the Karhunen–Loéve and cosine transform. *Proceedings of the Institution of Electrical Engineers*, Pt F, **128**, 359–360.

Gilbert, J., and Gilbert, L., 1988, *Elements of Modern Algebra* (Boston, Mass: Prindle, Weber, and Schmidt—Kent Publishing).

Hamidi, M., and Pearl, J., 1976, Comparison of the cosine and Fourier transforms of Markov-1 signals. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, **24**, 428–429.

Hou, H. S., 1987 A fast recursive algorithm for computing the discrete cosine transform. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, **35**, 1455–1461.

Jain, A. K., 1981, Image data compression: a review. *Proceedings of the Institute of Electrical and Electronics Engineers*, 349–389.

Makhoul, J., 1980, A fast cosine transform in one and two dimensions. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, **28**, 27–34.

Sun, M. T., Wu, L., and Liou, M. L., 1987, A concurrent architecture for VLSI implementation of discrete cosine transform. *I.E.E.E. Transactions on Circuits and Systems*, **34**, 8, 992–994.

Vetterli, M., and Ligtenberg, A., 1986, A discrete Fourier-cosine transform chip. *I.E.E.E. Journal of Selected Areas in Communications*, **4**, 49–61.

Volder, J. E., 1959, The CORDIC trigonometric computing technique. *I.R.E. Transactions on Electronic Computers*, **8**, 330–334.

Wold, E. H., and Despain, A. M., 1984, Pipeline and parallel-pipeline FFT processors for VLSI implementations. *I.E.E.E. Transactions on Computers*, **33**, 414–426.

ZELINSKI, R., and NOLL, P., 1977, Adaptive transform coding of speech signals. *I.E.E.E. Transactions on Acoustic, Speech and Signal Processing*, **25**, 299–309.

WU, JA-LING, and DUH, WEI-JOU, 1990, A novel concurrent architecture to implement discrete cosine transform based on index partitions. *The International Journal of Electronics*, to be published.