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

Novel concurrent architecture to implement the discrete cosine transform based on index partitions

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

Online Publication Date: 01 February 1990

To cite this Article Wu, Ja-Ling and Duh, Wei-Jou(1990)'Novel concurrent architecture to implement the discrete cosine transform based on index partitions',International Journal of Electronics,68:2,165 — 174
To link to this Article: DOI: 10.1080/00207219008921157
URL: http://dx.doi.org/10.1080/00207219008921157

INT. J. ELECTRONICS, 1990, VOL. 68, NO. 2, 165–174

Novel concurrent architecture to implement the discrete cosine transform based on index partitions

JA-LING WU† and WEI-JOU DUH†

In this paper a new concurrent architecture based on index partition and CORDIC techniques for implementing discrete cosine transforms (DCT) (with power of two length) is proposed. This architecture works basically in a serial-in parallel-out mode. In this newly proposed architecture, three stages of pipelining are applicable and the throughput rate is improved. Each processing element (PE) is basically a CORDIC processor with a fixed angle rotation and only $N/2$ PEs are required for computing an N-point DCT. Since each stage of the pipeline is nearly balanced, the concurrency of pipelining is explored as much as possible. The throughput rate of this architecture is $(N + 2)/NT$, where N is the transform length and T the system clock period. Therefore, in this architecture, the clock period can easily be pushed up to 50 ns in VLSI chips and the throughput rate would be 17·7 MHz for $N = 16$. Thus, this newly proposed architecture provides the posibility of real-time computations.

1. Introduction

The discrete cosine transform (DCT) was first proposed by Ahmed *et al.* (1974) and has become more and more important in recent years because its performance is closer to the optimal Karhunen–Loeve transform (KLT) than any other orthogonal transform (Haralick 1976, Madihally *et al.* 1978, Lee 1984). The DCT of length N is defined as (Ahmed *et al.* 1974)

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

where $n = 0, 1, 2, \ldots, N - 1$ and $\pi = 3{\cdot}14159$. Correspondingly, the inverse DCT (IDCT) is

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

where $k = 0, 1, 2, \ldots, N - 1$ and $c(n)$ is given by

$$\left.\begin{aligned} c(n) &= 1/\mathrm{sqrt}(2) \quad \text{for } n = 0 \\ &= 1 \qquad\qquad\ \ \text{otherwise} \end{aligned}\right\} \tag{3}$$

where sqrt(2) denotes the square root of 2.

Since the DCT can be used to compress a large amount of data with a small penalty it has been applied to various signal processing categories (e.g. image and speech coding). In order to meet the requirement of real-time processing, much attention has been given to the subject to real-time DCT processor design (Sun *et al.* 1987, Sikstrom *et al.* 1987, Vetterli and Ligtenberg 1986).

Received 30 June 1988; accepted 24 October 1988.

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

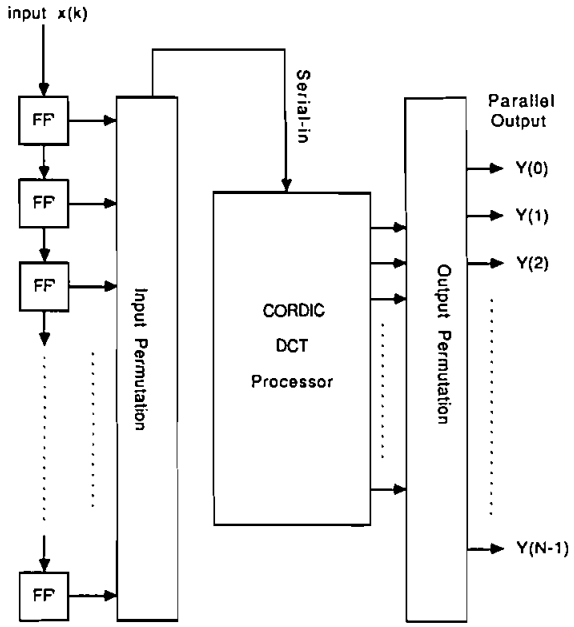

Figure 1. Block diagram of the CORDIC DCT architecture. FF' means data latch.

In this paper, a new concurrent architecture for implementing the DCT, with power of two length, is proposed. Owing to its simple and regular structures, this newly proposed architecture can achieve a very high throughput rate.

The organization of this paper is as follows. Section 2 presents a brief review of recent research on DCT processor design. Section 3 describes the new concurrent CORDIC DCT architecture. Finally, the conclusions and discussions are presented in § 4.

2. Review of DCT processor architectures

Most DCT processors are constructed by using distributed arithmetic, such as the DCT chip designed by Sun *et al.* (1987) and Sikstrom *et al.* (1987). An alternative using the discrete Fourier–cosine transformer (DFCT) is given by Vetterli and Ligtenberg (1986).

By using distributed arithmetic, one can construct specific hardware for computing the DCT with a very simple structure, as illustrated in Fig. 2 (Sun *et al.* 1987). The major parts of such an architecture are ROMs and accumulators. This architecture is of interest especially in its VLSI implementation because of the regularity and the high packing density of ROMs.

The DFCT architecture is based on the fast Fourier transform (FFT) algorithm (Vetterli and Ligtenberg 1986) and possesses butterfly-like structure. It can achieve better multiplicative complexity, but suffers from irregular data transfers. For completeness, a brief review of the distributed DCT architecture is given in the rest of this section.

In binary representation, $x(k)$, can be denoted as

$$x(k) = -x_0(k) + \sum_{r=1}^{W-1} x_r(k)2^{-r} \tag{4}$$

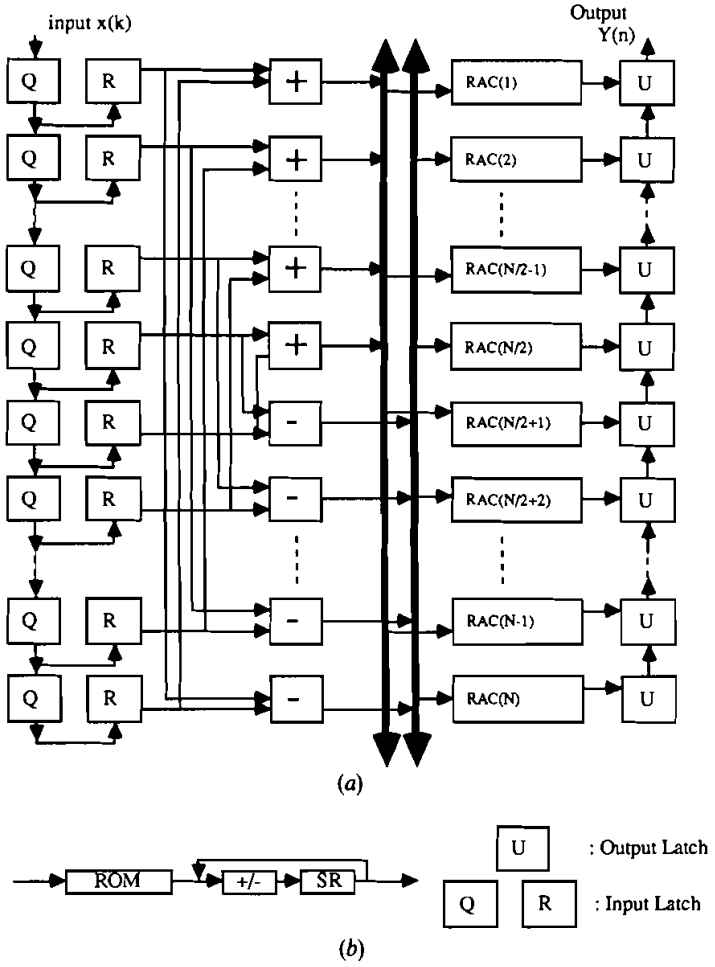

Figure 2. Distributed DCT processor: (*a*) system architecture; (*b*) RAC.

for $r = 0, 1, 2, \ldots, W-1$, where $x_r(k) = 0$ or 1. x_0 stands for the sign bit and W is the wordlength. Then (1) can be rewritten as

$$Y(n) = 2c(n)\Big/N \sum_{k=0}^{N-1}\left[-x_0(k) + \sum_{r=1}^{W-1} x_r(k)2^{-r}\right]\cos\left[\pi(2k+1)n/2N\right]$$

$$= 2c(n)\Big/N \sum_{k=0}^{N-1}\left[-x_0(k)\cos\left[\pi(2k+1)n/2N\right]\right.$$

$$\left. + \sum_{r=1}^{W-1}\sum_{N-1}^{N-1} x_r(k)\cos\left[\pi(2k+1)n/2N\right]2^{-r}\right] \tag{5}$$

Now define

$$F(r, n) = 2c(n)\Big/N \sum_{k=0}^{N-1} x_r(k)\cos\left[\pi(2k+1)n/2N\right] \tag{6 a}$$

for $r = 1, 2, \ldots, W - 1$, and

$$F(0, n) = 2c(n) \Big/ N \sum_{k=0}^{N-1} - x_0(k) \cos [\pi(2k + 1)n/2N] \tag{6 b}$$

From (5) and (6) it follows that

$$Y(n) = F(0, n) + \sum_{r=1}^{W-1} F(r, n)2^{-r} \tag{7}$$

The values of $F(r, n)$ are often taken from ROM tables and the computation of the DCT can now be implemented through table look-ups, shifts and adds. The complexities can be further reduced by using the symmetry properties of the DCT (Sun *et al.* 1987, Sikstrom *et al.* 1987); that is, for n even

$$Y(n) = \sum_{k=0}^{N/2-1} [x(k) + x(N - k - 1)] \cos [\pi(2k + 1)n/2N] \tag{8 a}$$

and for n odd

$$Y(n) = \sum_{k=0}^{N/2-1} [x(k) - x(N - k - 1)] \cos [\pi(2k + 1)n/2N] \tag{8 b}$$

The block diagram of the distributed DCT processor is shown in Fig. 2. The total hardware costs are as follows:

Memory	$N2^{N/2}W$ bits
Adders/subtractors	N
Accumulators	N

The total computing time is WM_A, where M_A is the system cycle time including the ROM accessing, one bit shifting and accumulations. In order to improve the performance further, a new architecture for computing the DCT is proposed in this paper.

3. Concurrent CORDIC DCT architecture

3.1. *CORDIC processors*

From the definition of the DCT given in (1), it follows that the transform kernels of the DCT, $\cos [2\pi(2k + 1)n/4N]$, are located at those $4N$ equally spaced points of the unit circle where N is the transform length. For ease of explanation a simple review of the CORDIC processors is given below.

A CORDIC processor is an example of the special-type processors proposed by Volder (1959). The CORDIC processor can perform coordinate rotations very easily and it can compute $\cos \theta$ and $\sin \theta$ (or $\cos (\pi/2 - \theta)$) concurrently. This means that CORDIC processors can compute two different cosine angles at the same time.

A point is located at the coordinate (X, Y) on a plane and X and Y are written as polar coordinates:

$$\left.\begin{aligned} X &= R \cos \theta \\ Y &= R \sin \theta \end{aligned}\right\} \tag{9}$$

where R is the distance between (X, Y) and the origin, and θ is the angle between the X-axis and the line from the origin to (X, Y). Suppose the vector (X, Y) rotates

through an angle θ'. The new coordinates then become

$$X' = R\cos(\theta + \theta') = R\cos\theta\cos\theta' - R\sin\theta\sin\theta'$$
$$= X\cos\theta' - Y\sin\theta' = \cos\theta'(X - Y\tan\theta') \quad (10a)$$

and

$$Y' = Y\cos\theta' + X\sin\theta' = \cos\theta'(Y + X\tan\theta') \quad (10b)$$

In general-purpose CORDIC processors θ' can be represented as a linear combination in terms of $a_i\theta_i$, where $a_i = 1$ or -1 and the θ_i are chosen as $\tan^{-1} 2^{-i}$. Thus, one can obtain the values of θ_i from ROMs, and only shifts and adds are required to compute X' and Y'. A block diagram of the general CORDIC processor is shown in Fig. 3.

It is observed that when CORDIC processors are used in the implementation of the DCT, fixed-angle rotations only need be computed. This means that higher computational speed can be achieved with minimal special hardware (Despain 1979).

3.2. *Index partition*

In order to achieve high-speed computation of the DCT, a specific index partition is required. In what follows we assume the transform length N equals 2^m, m being an integer, and we denote by k and n the indices of the input and output,

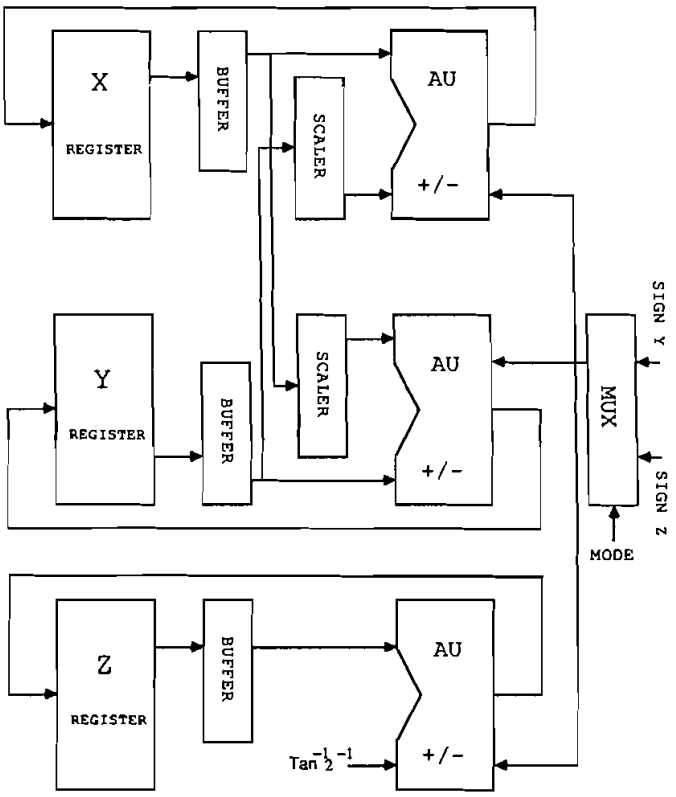

Figure 3. Block diagram of a CORDIC processor.

respectively. The output index n is now partitioned into m sets according to the formula given below. (For simplicity, and without loss of generality, the normalization factor $2/N$ in (1) is ignored in this index partition.)

Let

$$n = 2^{L+1}p + 2^L \tag{11}$$

where $L = 0\ 1, 2, \ldots, m-1$ and $p = 0, 1, \ldots, 2^{m-L-1} - 1$, then

$$Y(0) = c(0) \sum_{k=0}^{N-1} x(k) = \sum_{k=0}^{N-1} x(k) \cos [\pi/4] \tag{12a}$$

and

$$Y(n) = \sum_{k=0}^{N-1} x(k) \cos [\pi(2k+1)(2p+1)/2^{i+1}] \tag{12b}$$

for n not equal to zero and $L = m - i$, $1 \leqslant i \leqslant m$.

Since the term given in (12a) is of the same form as that of $L = m - 1$ in (12b), (12a) can be combined with (12b). From equation (12) it is clear that the output index n can be partitioned into m sets corresponding to $L = 0, 1, \ldots, m-1$, respectively. Why this particular partition? Because for each n and a given L, only a few cosine values of some fixed angles are required in computing $Y(n)$. Hence, we call the index partition given in (11) the 'angle partition'. Figure 4 illustrates the basic idea of such a partition.

Consider, for example, the case of $N = 8$. The sets associated with $L = 0, 1, 2$ are $\{1, 3, 5, 7\}$, $\{2, 6\}$ and $\{4, 0\}$, respectively. In another interpretation, the sets correspond to $(n, N) = 2^0$, $(n, N) = 2^1$, $(n, N) = 2^2$, respectively, where (n, N) denotes the greatest common divisor (g.c.d.) of n and N. Since these sets are mutually disjoint it is evident that (11) forms a complete index partition on n.

From the symmetric property of the DCT, as in (8), another decomposition on n can be applied. From (8) the output indices can be grouped in pairs. The new decomposition is defined as follows:

$$\begin{aligned} S(n) &= \{Y(0), Y(N/2)\} \quad \text{for} \quad n = N/2 \\ &= \{Y(n), Y(N-n)\} \quad \text{for} \quad n = 1, 2, \ldots, N/2 - 1 \end{aligned} \tag{13}$$

There are $N/2$ pairs, $S(n)$, in which outputs with the indices in the same pair can be computed in the same CORDIC processor concurrently. This is so because the

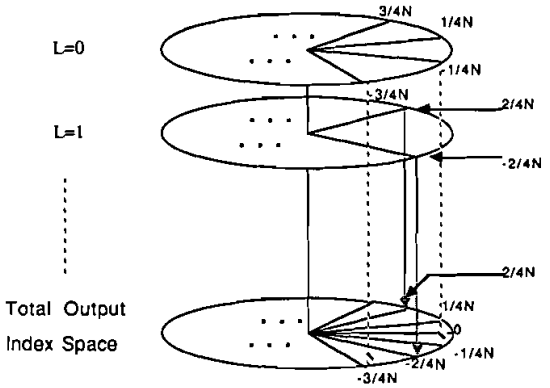

Figure 4. Angle partition.

CORDIC processor can compute sine and cosine functions simultaneously, as indicated in (14). However, each $S(n)$ is physically an accumulator pair:

$$Y(n) = \sum_{k=0}^{N-1} x(k) \cos [\pi(2k + 1)n/2N] \tag{14a}$$

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

$$= \sum_{k=0}^{N-1} (-1)^k x(k) \sin [\pi(2k + 1)n/2N] \tag{14b}$$

For simplicity, we denote $\cos [i\pi/2N]$ by C^i and $\sin [i\pi/2N]$ by S^i. In the example of $N = 16$, for each $S(n)$ with n in $\{1, 3, 5, 7\}$, only $\{C^1, C^3, C^5, C^7, S^1, S^3, S^5, S^7\}$ are used; for each $S(n)$ with n in $\{2, 6\}$, only $\{C^2, C^6, S^2, S^6\}$ are used, and so on. This leads to the concurrent CORDIC DCT architecture shown in Fig. 5. Note that each CORDIC processor in the proposed architecture performs only a fixed-angle rotation. That is, it can be implemented efficiently and easily by using the rational-number approximation techniques given in Despain (1979). The interconnection networks (INs) transmit the intermediate results of each CORDIC processor to appropriate output accumulators (Acc.) to produce the correct results. For completeness, the control and routing strategies of the INs will be described in the following section.

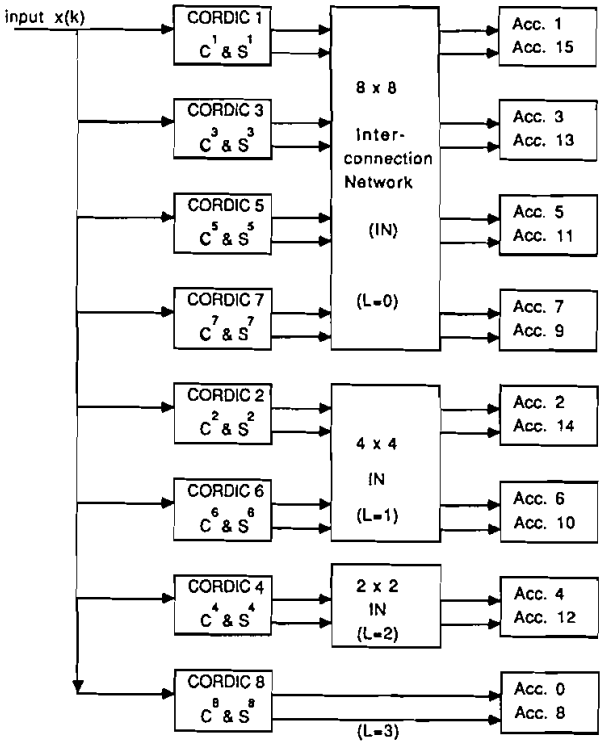

Figure 5. 16-Point concurrent CORDIC DCT architecture.

3.3. *Control and routing strategies of the INs*

According to the partitions mentioned above, a simple and direct architecture can be constructed. In such a configuration, there are m sets of output accumulators corresponding to $L = 0, 1, \ldots, m-1$, respectively. Each set comprises CORDIC processors, interconnection networks and accumulators. Thus there are three stages in this configuration, and pipeline design strategies can be applied to achieve a higher throughput rate.

As previously mentioned, the CORDIC processors can compute sine and cosine functions simultaneously. We then define a binary operator '$\&_N$' over the set $\{1, 3, 5, \ldots, N/2-1\}$ to indicate in which CORDIC processor the value C^i and S^i should be computed.

Definition 1

'$\&_N$' is a binary operator over the set $\{1, 3, 5, \ldots, N/2-1\}$ and its definition is given by

$$\left.\begin{aligned} x \,\&_N\, y &= N - (xy \bmod N) && \text{if } (xy \bmod N) > N/2 \\ &= xy \bmod N && \text{otherwise} \end{aligned}\right\} \tag{15}$$

With the aid of '$\&_N$', the control signals of the IN can be generated as follows:

(*a*) For input index k, the output of the ith CORDIC processor must route to the accumulator set that is numbered as $(2k+1)^{-1} \,\&_N\, i$, where x^{-1} is the inverse element of x under the operator '$\&_N$'. *Example*: Consider the case of $N = 16$ and if the input is $x(2)$, i.e. input index is 2, the destination of the 3rd CORDIC processor is 7 ($5^{-1} \,\&_{16}\, 3$). That is, the output of the 3rd CORDIC processor must route to $S(7)$, i.e. the accumulators 7 and 9.

(*b*) Each CORDIC processor has two outputs, $x(k)C^i$ and $x(k)S^i$, which will be routed to two accumulators, say 'Acc.n' and 'Acc.$N-n$', respectively. If $(2k+1)n \bmod 2N$ is greater than $3N/2$ or smaller than $N/2$, then the cosine output of the CORDIC processor will be routed to 'Acc.n' and sine output to 'Acc.$N-n$'; otherwise, we change the routing paths. *Example*: Consider the same case as in (*a*): $k = 2$, $n = 7$ and $N - n = 9$. According to the rule state above, the consine output of the 3rd CORDIC processor should route to 'Acc.9' and the sine output to 'Acc.7'.

The add/subtract control signals of the accumulators, i.e. the signs of C^i and S^i, can be generated by the following rules:

(*c*) Using the same notation as given in (*b*), the add/subtract control of the accumulator n is determined by the value of $(2k+1)n \bmod 4N$. If it is greater than $3N$ or smaller than N, then addition is performed; otherwise, subtraction is performed.

These control signals can be generated by a simple computer program and preloaded into the controllers of the INs. An effective cell-based IN has been proposed by Wu and Lin (1985), which can be used to realize this new architecture directly.

3.4. *Design example*

Consider the case of $N = 16$ and take $x(2)$ as the input. First, find the matching between the CORDIC processors and the accumulator pairs according to the

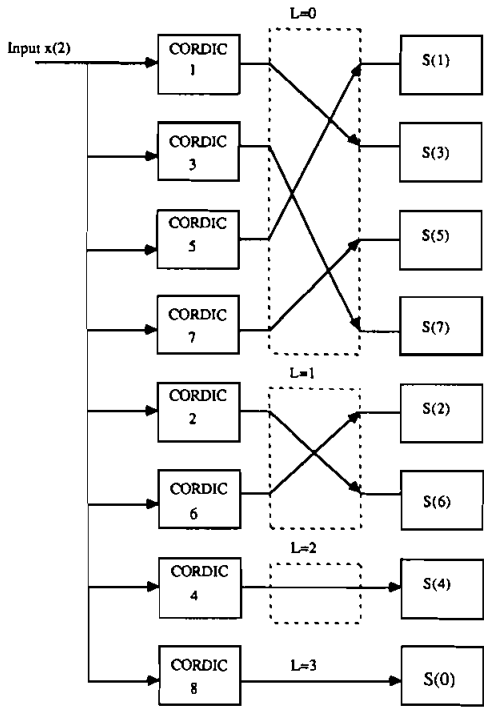

Figure 6. Configuration of INs for $x(2)$.

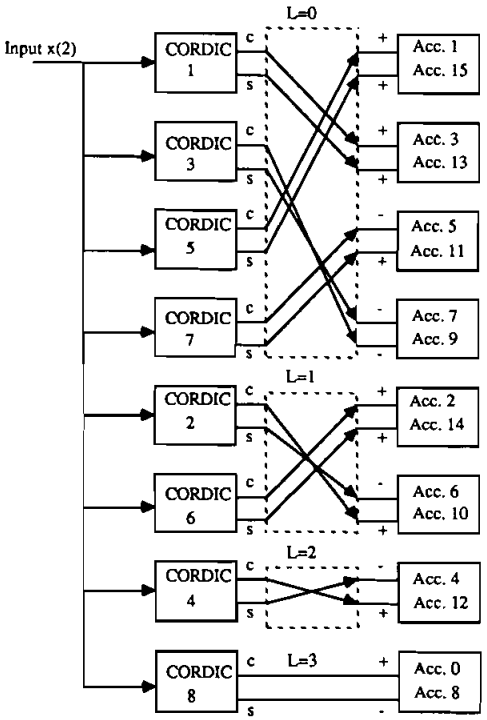

Figure 7. Detailed control patterns of INs for $x(2)$.

method mentioned in (*a*). Since the inverse element of 5 ($2 \times 2 + 1$) under '$\&_{16}$' is 3. The routing scheme should be as shown in Fig. 6.

Second, determine the routing paths of $x(2)C^i$ and $x(2)S^i$ as in (*b*). Finally, the signs of $x(2)C^i$ and $x(2)S^i$ are determined by using the method mentioned in (*c*). The final routing of the INs is shown in Fig. 7.

For another input one can follow the same procedure to find all the control signals and complete the design.

4. Discussion and conclusions

In this paper we proposed a new concurrent architecture to implement the DCT using CORDIC processors. The architecture is in serial-in parallel-out mode. It achieves a high throughput rate, because N data are all computed concurrently and three stages of pipelining are applicable. Since the number of the stages of the pipelining is a constant, the throughput rate increases with N. Furthermore, the concurrent CORDIC DCT receives its input in natural order.

The throughput rate of this architecture is $(N + 2)/N \cdot T$, since three stages of pipelining are applicable. If the system clock period T is 50 ns the throughput rate is about 16 MHz for $N = 8$ and 17·7 MHz for $N = 16$. It is clear that this newly proposed architecture has the capability of real-time computation.

References

Ahmed, N., Natarajan, T., and Rao, K. R., 1974, Discrete cosine transform. *I.E.E.E. Transactions on Computers*, **23**, 90–94.

Despain, A. M., 1979, Very fast Fourier transform algorithms: Hardware for implementation. *I.E.E.E. Transactions on Computers*, **28**, 333–341.

Haralick, R. M., 1976, A storage efficient way to implement the discrete cosine transform. *I.E.E.E. Transactions on Computers*, **25**, 764–765.

Lee, B. G., 1984, A new algorithm to compute the Discrete cosine transform. *I.E.E.E. Transactions on Acoustics, Speech and Signal Processing*, **32**, 1243–1247.

Narasimha, M. J., and Peterson, A. M., 1978, On the computation of the discrete cosine transform. *I.E.E.E. Transactions on Communications*, **26**, 934–936.

Sikstrom, B., Wanhammar, L., Afghahi, M., and Pencz, J., 1987, A high speed 2-D discrete cosine transform chip. *Integration, The VLSI Journal*, **5**, 159–169.

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**, 992–994.

Vetterli, M., and Ligtenberg, A., 1986, A discrete Fourier-cosine transform chip. *I.E.E.E. Journal on 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.

Wu, Ja-Ling, and Lin, Teng-Pin, 1985, A new cell-based interconnection network. *International Journal of Electronics*, **59**, 375–382.