

High Throughput CORDIC-Based Systolic Array Design for the Discrete Cosine Transform

Jue-Hsuan Hsiao, Liang-Gee Chen, Tzi-Dar Chiueh and Chun-Te Chen

Department of Electrical Engineering, Rm. 332

National Taiwan University, Taipei,Taiwan, 10764 R.O.C

ABSTRACT

In this paper, we propose a modified Malvar algorithm that can transfer the Discrete Hartley Transform (DHT) results to the Discrete Cosine Transform (DCT) by using one additional CORDIC computing stage. A fast CORDIC-based systolic array is designed with four attractive features, including: 1) the single/double data folding feature; 2) the constructive feature; 3) the co-computing feature; and 4) the redundant path computation. Due to the existence of redundant path, the proposed design also have capability of error detection.

1. INTRODUCTION

Due to the satisfactory performance close to the statistically optimal Karhunen-Loeve transform, the Discrete Cosine Transform (DCT) is widely used in block signal coding for a wide class of signals. Fast algorithms for the DCT are, therefore, of significant practical interest. These algorithms can broadly be classified into three groups: 1) Direct Factorisation of the DCT matrix; 2) indirect computation through fast Fourier transform [FFT] or through Discrete Hartley transform [DHT] [1]; and 3) algorithms based on complexity theory.

However, by using direct or indirect algorithm to construct systolic array for the DCT transform has two drawbacks: 1) high memory bandwidth and 2) the difficulty of butterfly shuffling between stages. On the contrary, the Discrete Hartley Transform was lately introduced as a new tool for the analysis, design and implementation of digital signal processing algorithms and systems. It is strictly symmetrical concerning the transform and its inverse. Applied it to real signals, it is faster than a real Fourier transform, especially in the case of the inverse transform. The speed of operation for a fast convolution can thus be increased. So, the DHT transform has been considered as an alternative to the FFT for spectral analysis and fast convolution of real data.

In this paper, we will present a new fast, parallel and high throughput systolic array design for the DCT computation, in which a DHT transform based on CORDIC algorithm is performed first, and then the result is passed through Hartley-to-Cosine transform.

2. CORDIC ALGORITHM

The CORDIC algorithm was first introduced by Volder [2] for the computation of trigonometric functions, multiplication, division, and data type conversion, and was generalized to hyperbolic functions by Walther [3]. The major advantage of this algorithm is given by the fact that it can be realized as sequence of additions/subtractions and shift operations. Since these operations are very well suited for VLSI realization, the algorithm has attracted a lot of attention in the recent times. For a detailed description of the CORDIC algorithm the readers are referred to Volder's and Walther's original works.

In the DHT computation, the elementary trigonometric functions such as sine and cosine functions are involved. In order to facilitate these operations, the CORDIC will operate

in the circular coordinate system. Fig. 1 shows the rotation function computed by the CORDIC algorithm in this system. By setting $x_n = A_n$, $y_n = B_n$ and $Z_{mn} = \theta_{mn} = (2\pi mn/N)$, one can obtain $B_n * \cos(2\pi mn/N) + A_n * \sin(2\pi mn/N)$ from $\text{cordic_Y}(m,n)$ and $A_n * \cos(2\pi mn/N) - B_n * \sin(2\pi mn/N)$ from $\text{cordic_X}(m,n)$ concurrency. By using CORDIC technique, these two kernels can be simultaneously implemented in the same hardware.

3. THE COMPUTATION OF THE DCT TRANSFORM THROUGH DHT COMPUTATION

The one-dimensional DCT of N points, $DCT(N)$, is defined as

$$C(m) = \frac{1}{N} e(m) \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{2N}(2n+1)m\right], \quad 0 \leq m \leq N-1 \quad \dots(1)$$

where $e(m) = \frac{1}{\sqrt{2}}$ if $m=0$; otherwise $e(m) = 1$. Since the factor $\frac{1}{N}e(m)$ results only in a slight modification of $C(m)$, it is sufficient to consider the DCT-like equation

$$C(m) = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{2N}(2n+1)m\right], \quad 0 \leq m \leq N-1 \quad \dots(2)$$

There are various schemes for computing 1-D DCT [1-3]. Since the number of multiplications of the Winograd-Hartley transform is minimum, the DHT transform has adopted to calculate the DCT transform. In [4], a new data sequence h_n is defined as

$$\begin{aligned} h_n &= x_{2n} & n &= 0, 1, \dots, N/2-1 \\ &= x_{2N-2n-1} & n &= N/2, \dots, N-1 \end{aligned} \quad \dots(3)$$

and the DHT of h_n is defined as

$$H_m = \sum_{n=0}^{N-1} h_n \text{cas}\left(\frac{2\pi}{N}mn\right) \quad \dots(4)$$

where $\text{cas}(\cdot) = \cos(\cdot) + \sin(\cdot)$. Combining Eq. (3) and (4), we obtain

$$H_m = \sum_{\text{even}} x_n \text{cas}\left(\frac{m\pi}{N}\right) + \sum_{\text{odd}} x_n \text{cas}\left(-\frac{m\pi}{N}(n+1)\right) \quad \dots(5)$$

By using the identity $\text{cas}(x)\text{cas}(y) = \cos(x-y) + \sin(x+y)$, Malvar has shown that

$$C(m) = \frac{1}{2} [H_m \text{cas}\left(-\frac{m\pi}{2N}\right) + H_{N-m} \text{cas}\left(\frac{m\pi}{2N}\right)] \quad \dots(6)$$

and he has also proposed the following fast and parallel algorithm for computing the DCT transform through the DHT transform

$$\begin{bmatrix} C(m) \\ C(N-m) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \text{cas}(-m\pi/2N) & \text{cas}(m\pi/2N) \\ \text{cas}((N-m)\pi/2N) & \text{cas}(-(N-m)\pi/2N) \end{bmatrix} \begin{bmatrix} H_m \\ H_{N-m} \end{bmatrix} \quad \dots(7)$$

where $m = 0, 1, 2, \dots, N/2$ and $H_0 = H_N$. For our purpose, we define a $\text{css}(\cdot) = \cos(\cdot)\sin(\cdot)$ function that has the following properties :

$$\begin{aligned} \text{cas}(2\pi - \frac{\pi}{2N}m) &= \text{css}(\frac{\pi}{2N}m), \quad \text{cas}(\frac{\pi}{2} - \frac{\pi}{2N}m) = \text{cas}(\frac{\pi}{2N}m), \text{ and} \\ \text{cas}(\frac{\pi}{2N}m - \frac{\pi}{2}) &= -\text{css}(\frac{\pi}{2N}m). \end{aligned}$$

Moreover, we use these identities to simplify Eq. (7) as below

$$\begin{bmatrix} C(m) \\ C(N-m) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \text{css}(m\pi/2N) & \text{cas}(m\pi/2N) \\ \text{cas}(m\pi/2N) & -\text{css}(m\pi/2N) \end{bmatrix} \begin{bmatrix} H_m \\ H_{N-m} \end{bmatrix} \quad \dots(8)$$

This formula indicates that let $X_n = Y_n = H_m$ and $Z_{mn} = (m\pi/2N)$ at one CORDIC processor input ports, $H_m \text{cas}(m\pi/2N)$ and $H_m \text{css}(m\pi/2N)$ will be obtained from output ports. In the same manner, let $X_n = Y_n = H_{N-m}$ and $Z_{mn} = -(m\pi/2N)$ at another CORDIC processor input ports, $H_{N-m} \text{cas}(m\pi/2N)$ and $H_{N-m} \text{css}(m\pi/2N)$ will be obtained from this processor. After this, one subtraction ($H_m \text{cas}(m\pi/2N) - H_{N-m} \text{css}(m\pi/2N)$)

and one shift right operation are proceeded to get $C(N-m)$ result. Meanwhile, one addition ($H_m \text{ cas}(m\pi/2N) + H_{N-m} \text{ css}(m\pi/2N)$) and one shift right operation are also proceeded to get $C(m)$ result. Therefore, an approach to high throughput DCT transform through the DHT transform will be achieved by using the CORDIC processor and Eq. 8.

4. A FAST SYSTOLIC ARRAY DESIGN FOR DHT

The Discrete Hartley Transform (DHT) can be written as

$$H_m = \sum_{n=0}^{N-1} h_n \text{cas}\left(\frac{2\pi}{N}mn\right)$$

where $m=0, 1, 2, \dots, N-1$ and $\text{cas}(\cdot) = \cos(\cdot) + \sin(\cdot)$ (9)
There are four properties derived for 1-D DHT transform in [4]. They are: 1) the single/double data folding feature; 2) the constructive feature; 3) the co-compute between H_m and H_{N-m} ; and 4) the redundant path computation. We will briefly introduce these features in the following paragraphs.

4a. The Data Folding

When $N=2^x$ and $x \geq 1$, Eq. (9) can be folded as follows :

$$\begin{aligned} H_m &= \sum_{n=0}^{N/2-1} [(h_n + (-1)^m h_{n+N/2}) \cos\left(\frac{2\pi}{N}mn\right)] \\ &\quad + \sum_{n=0}^{N/2-1} [(h_n + (-1)^m h_{n+N/2}) \sin\left(\frac{2\pi}{N}mn\right)] \\ &= \sum_{n=0}^{N/2-1} [(h_n + (-1)^m h_{n+N/2}) \text{cas}\left(\frac{2\pi}{N}mn\right)] \\ &= \sum_{n=0}^{N/2-1} [(x_{2n} + (-1)^m x_{N-2n-1}) \text{cas}\left(\frac{2\pi}{N}mn\right)] \end{aligned} \quad \dots \dots \dots (10)$$

From Eq. 10, one can also notice that the DHT of N points has been folded into $N/2$ points DHT to reduce multiplication and addition counts. This process is so called Single Data Folding (SDF). On the other hand, when $N=4^x$ and $x \geq 1$, Eq. (9) can also be folded as follows :

$$\begin{aligned} H_m &= \sum_{n=0}^{N/4-1} [A_{m,n} \sin\left(\frac{2\pi}{N}mn\right) + B_{m,n} \cos\left(\frac{2\pi}{N}mn\right)] \\ A_{m,n} &= [(h_n + (-1)^m h_{n+N/2}) + (h_{n+N/4} + (-1)^m h_{n+3N/4})] \\ &= [(x_{2n} + (-1)^m x_{N-2n-1}) + (x_{2n+N/2} + (-1)^m x_{N/2-2n-1})], \end{aligned}$$

when m mode 4 = 0,3

$$\begin{aligned} &= [(h_n + (-1)^m h_{n+N/2}) - (h_{n+N/4} + (-1)^m h_{n+3N/4})] \\ &= [(x_{2n} + (-1)^m x_{N-2n-1}) - (x_{2n+N/2} + (-1)^m x_{N/2-2n-1})], \end{aligned}$$

when m mode 4 = 1,2.

$$\begin{aligned} B_{m,n} &= [(h_n + (-1)^m h_{n+N/2}) + (h_{n+N/4} + (-1)^m h_{n+3N/4})] \\ &= [(x_{2n} + (-1)^m x_{N-2n-1}) + (x_{2n+N/2} + (-1)^m x_{N/2-2n-1})], \end{aligned}$$

when m mode 4 = 0,1

$$\begin{aligned} &= [(h_n + (-1)^m h_{n+N/2}) - (h_{n+N/4} + (-1)^m h_{n+3N/4})] \\ &= [(x_{2n} + (-1)^m x_{N-2n-1}) - (x_{2n+N/2} + (-1)^m x_{N/2-2n-1})], \end{aligned}$$

when m mode 4 = 2,3.

$$C = N/4. \quad \dots \dots \dots (11)$$

Eq. 11 also notice that the DHT of N points have been folded into $N/4$ points DHT to greatly reduce multiplication and addition counts. This process is so called Double Data Folding (DDF).

4b. The Constructive Feature

The folded H_k can be decomposed into four sub-summations :

$$\begin{aligned} H_k &= \left\{ \sum_{n=0}^{C/4-1} [A_{k,4n} \sin\left(\frac{2\pi}{N}k(4n)\right) + B_{k,4n} \cos\left(\frac{2\pi}{N}k(4n)\right)] + \right. \\ &\quad \left. \sum_{n=0}^{C/4-1} [A_{k,4n+2} \sin\left(\frac{2\pi}{N}k(4n+2)\right) + B_{k,4n+2} \cos\left(\frac{2\pi}{N}k(4n+2)\right)] \right\} + \\ &\quad \left\{ \sum_{n=0}^{C/4-1} [A_{k,4n+1} \sin\left(\frac{2\pi}{N}k(4n+1)\right) + B_{k,4n+1} \cos\left(\frac{2\pi}{N}k(4n+1)\right)] \right\} + \end{aligned}$$

$$\sum_{n=0}^{C/4-1} [A_{k,4n+3} \sin\left(\frac{2\pi}{N}k(4n+3)\right) + B_{k,4n+3} \cos\left(\frac{2\pi}{N}k(4n+3)\right)] \} \quad \dots \dots \dots (12)$$

and the following identities

$$\begin{aligned} \text{cordic_Y}(m, n) &= \text{cordic_CO} - Y(m, n) \\ &= A_{m,n} \sin\left(\frac{2\pi}{N}mn\right) + B_{m,n} \cos\left(\frac{2\pi}{N}mn\right) \text{ and} \\ \text{cordic_X}(m, n) &= \text{cordic_CO} - Y(m, n) \\ &= A_{m,n} \cos\left(\frac{2\pi}{N}mn\right) - B_{m,n} \sin\left(\frac{2\pi}{N}mn\right) \end{aligned}$$

can be used in H_k to rewrite Eq. (12)

$$H_m = \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n) + \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2) \right\} + \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+1) + \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+3) \right\} \dots \dots \dots (12a)$$

$$H_{m+N/2} = \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n) + \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2) \right\} - \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+1) + \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+3) \right\} \dots \dots \dots (12b)$$

$$H_{m+N/4} = \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n) - \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2) \right\} + \left\{ \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+1) - \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+3) \right\} \dots \dots \dots (12c)$$

$$H_{m+3N/4} = \left\{ \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n) - \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2) \right\} - \left\{ \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+1) - \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+3) \right\} \dots \dots \dots (12d)$$

Eqs. (12a-12d) indicate that six constructive parts, $\sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n)$, $\sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2)$, $\sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+1)$, $\sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+3)$, $\sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+1)$ and $\sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+3)$ are used to construct H_m , $H_{m+N/4}$, $H_{m+N/2}$ and $H_{m+3N/4}$. As the pair of $\text{cordic_Y}(m, 4n+1)$ and $\text{cordic_CO-Y}(m, 4n+1)$ and the pair of $\text{cordic_Y}(m, 4n+3)$ and $\text{cordic_CO-Y}(m, 4n+3)$ are mutual co_result, they come out of the same computing hardware concurrently. Hence, only four parts of computing hardware are needed to construct them. Obviously, an four times throughput DHT transform will be achieved by using this constructive feature.

4c. The Co-computing between H_m and H_{N-m}

On the other hand, H_{N-m} can be shown as follows :

$$\begin{aligned} H_{N-m} &= \sum_{n=0}^{C/4-1} [A_{N-m,4n} \cos\left(\frac{2\pi}{N}m(4n)\right) - B_{N-m,4n} \sin\left(\frac{2\pi}{N}m(4n)\right)] + \\ &\quad \sum_{n=0}^{C/4-1} [A_{N-m,4n+2} \cos\left(\frac{2\pi}{N}m(4n+2)\right) - B_{N-m,4n+2} \sin\left(\frac{2\pi}{N}m(4n+2)\right)] + \\ &\quad \sum_{n=0}^{C/4-1} [A_{N-m,4n+1} \cos\left(\frac{2\pi}{N}m(4n+1)\right) - B_{N-m,4n+1} \sin\left(\frac{2\pi}{N}m(4n+1)\right)] + \\ &\quad \sum_{n=0}^{C/4-1} [A_{N-m,4n+3} \cos\left(\frac{2\pi}{N}m(4n+3)\right) - B_{N-m,4n+3} \sin\left(\frac{2\pi}{N}m(4n+3)\right)] \end{aligned} \quad \dots \dots \dots (13a)$$

By using the CORDIC technology to compute H_m , and we can find that

$$\begin{aligned} H_{N-m} &= \left\{ \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n) + \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+2) \right\} + \\ &\quad \left\{ \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+1) + \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+3) \right\} \end{aligned} \quad \dots \dots \dots (13b)$$

This equation indicates that H_{N-m} can be constructed from the result/co-result of H_m in the same hardware. In the same way, we can find that H_{N-m} , $H_{N-(m+3N/4)}$, $H_{N-(m+N/4)}$ and $H_{N-(m+N/2)}$ can be constructed by the following six constructive parts--

$$\begin{aligned} &\sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n), \sum_{n=0}^{C/4-1} \text{cordic_Y}(m, 4n+2), \\ &\sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n), \sum_{n=0}^{C/4-1} \text{cordic_CO} - Y(m, 4n+2), \end{aligned}$$

$\sum_{n=0}^{C/4-1}$ cordic_Y(m, 4n+1), and $\sum_{n=0}^{C/4-1}$ cordic_Y(m, 4n+3). And these six parts are also existing when we construct H_m , $H_{m+N/4}$, $H_{m+N/2}$ and $H_{m+3N/4}$. Thus, no additional computing efforts are required to construct them. It means that the throughput can be improved by eight times.

5. THE PROPOSED SYSTOLIC ARRAY

As noted in Eq. (8), each C(m) can be realized by the construction of CORDIC computing terms by data folded and modified Malvar calculation. There are totally four tasks to achieve the DCT, including data folding, DHT kernel computing by CORDIC processor, DHT constructing and DHT-to-DCT conversion. Four different cells are proposed to do these computations. First, the SDF(DDF) cell is designed to implement the data folding task. Fig. 2 shows the inner structure of SDF and DDF. Second, the CORDIC processor is developed to compute the DHT transform. The structure of the CORDIC processor is shown in Fig. 3. Third, a CONStructure cell is shown in Fig. 4. One CONStructure cell guided by Eqs. (12a-12d) can be used to construct H_m , $H_{m+N/4}$, $H_{m+N/2}$ and $H_{m+3N/4}$ and another cell guided by Eqs. (13a-13b) can be used to construct H_{N-m} , $H_{N-(m+N/4)}$, $H_{N-(m+N/2)}$, and $H_{N-(m+3N/4)}$. Finally, Fig. 5 shows the H-to-C cell designed for completing DHT-to-DCT conversion.

By using these four cells types and their features, a novel systolic array for the DCT can be obtained. Fig. 6 shows the proposed architecture with N= 32. Since N=4*4*x, the DDF cell is used to do single data folding before CORDIC computing and then followed by the CONStructure cells to get eight DHT transform results. These results are sent to H-to-C cells to obtain the exact DCT results.

The data flow of the proposed systolic array for the DCT computation are shown in Fig. 7. At clock 1, every DDF unit in stage 1 calculates A_{mn} and B_{mn} , and then the results are sent to the following CORDIC unit in the next stage for CORDIC_X(m,n) and CORDIC_Y(m,n) calculation at clock 2. Meanwhile, new DDF computing is going on. The arrow points out the direction of the data flow. After the clock 4, the DHT transform results of H_0 , H_8 , H_{16} and H_{24} are obtained from the CONStructure cell-1 and another four DHT transform results of H_0 , H_{24} , H_{16} and H_8 are obtained from the CONStructure cell-2. From this, another eight DHT transform results per clock will be achieved. Therefore, this proposed systolic array can get eight DCT results per clock after the latency. In other words, the proposed array can achieve eight times throughput compared with other previous works. The latency consists of four parts: one data folding clock, stage clocks, one result constructive clock and one H-to-C conversion clock. Incidentally, the stage clocks for computing each constructive part of H_m can be represented as:

stage clocks = N/(data folded factor * constructive factor)
where data folded factor = 2, for single data folding and
= 4, for double data folding.

$$\text{constructive factor} = 4$$

Fig. 8 indicates an addressing method on four-bank memory that is shown in Fig. 6. According the data flow in Fig. 7, the data introduce time is 5 clocks, and this time can be formulated as $(N/8)+1$.

Because the redundant path computations exists, an error detector has been design for the hardware failure detecting. Fig. 9 shows its inner structure and illustrates the

hardware error occurrence on $C(N-m) \neq C(m)$, when $m=0$ and $m=N/8$.

6. THE ANALYSIS OF COMPLEXITY

Suppose a conventional CORDIC processor requires $T_{\text{CORDIC}} = nT_c$ time to complete the computation, here n is the number of CORDIC iterations, $T_c = T_{\text{shift}} + T_{\text{add}}$ is the time of one CORDIC iteration and $T_{\text{shift}} + T_{\text{add}}$ are the shift and addition operations, respectively. Therefore, every CORDIC unit needs $T_{\text{CORDIC}} + T_{\text{add}}$ to complete the computation of $h_n \cos(\theta_{mn}) + h_n \sin(\theta_{mn})$ and $h_n \cos(\theta_{mn}) - h_n \sin(\theta_{mn})$.

In order to enhance the throughput of the DCT, the pipelined design will be considered to meet this requirement. The performances of proposed systolic array are listed in Table 1. The throughput rate is defined by the number of the DCT transformation in one clock. In our proposed systolic array, the latency can be formulated as $(N/2^4)*3$ [$(N/4^4)*3$] clocks for the SDF [DDF] mode, and the throughput rate is eight DCT transforms per clock. The turnaround time represents the total clocks needed to complete the whole DCT. The data introduce time indicates the time between two N data sequences. Another measurement, shown in the last term, is the throughput of the CORDIC unit. It can be defined as follows:

$$\text{Throughput of CORDIC} = \frac{N \text{ DCT}}{(\text{Data introduce time})(\text{Number of CORDIC used})}$$

The unit of the last term is DCT / cordic-clock. In our proposal, the throughput of the CORDIC in the SDF [DDF] mode is $16N/(N^2+16N+64)$ [$32N/(N^2+24N+128)$].

7. CONCLUSION

In this paper, we have derived a modified Malvar algorithm to meet the CORDIC based requirements and used it to get the DCT results via DHT computation. The transforming is only to permute the input sequences and append one CORDIC stage to the DHT systolic array. Thus, a high throughput and cost effective architecture has been designed for the 1-D DCT. Almost eight times performance can be obtained, while much cheaper hardware is implemented. Moreover, this design can provide error detection capability. Due to its regularity and simplicity, the proposed architecture will be very suitable for VLSI implementation.

8. REFERENCES

- [1] H. Malvar, "Fast computation of discrete cosine transform through fast Hartley transform," Electronic Letters, 1986, Vol. 22, No. 7, pp.352-353.
- [2] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Comput., Vol. EC-8, pp. 330-334, Sept. 1959.
- [3] J. S. Walther, "A unified algorithm for elementary functions," in AFIPS Spring Joint Comput. Conf., 1971, pp.379-385.
- [4] J. H. Hsiao, L. G. Chen, T. D. Chiueh and C. T. Chen, "Novel systolic array design for the discrete Hartley transform with high throughput rate," Proc. of IEEE International Symposium on Circuits and Systems, Chicago,Illinois, pp.1567-1570, May, 1993.

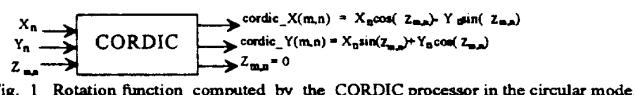


Fig. 1 Rotation function computed by the CORDIC processor in the circular mode

TABLE I. The performances of proposed systolic array

	Latency	Throughput rate	Turnaround time	CORDIC cells
SDF	(N/8)+3	8	(N/4)+3	(N/2)+4
DDF	(N/16)+3	8	(3N/16)+3	(N/4)+4
	SDF cells	DDF cells	Data introduce time	Throughput of CORDIC (1)
SDF	N/2	0	(N/8)+1	$16N(N^2+16N+64)$
DDF	0	N/4	(N/8)+1	$32N(N^2+24N+128)$

(1) : the unit is DCT/cordic-clock.

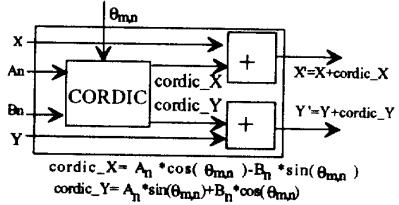


Fig. 3 The inner structure of CORDIC processor

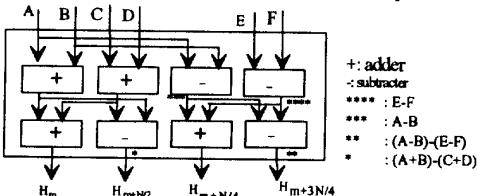


Fig. 4 The CONSTRUCTURE cell.

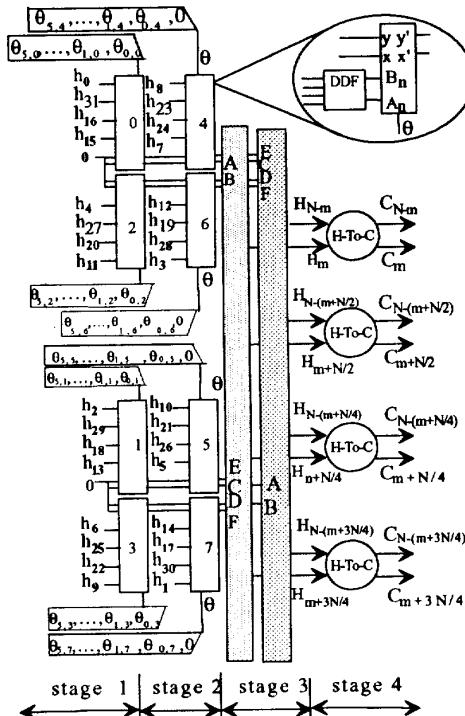
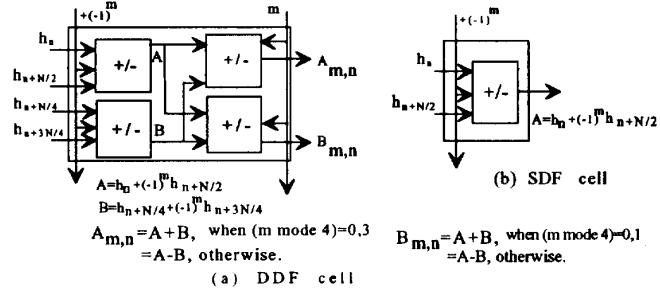
Fig. 6 A novel systolic array designed for DCT transform, when $N=32$. Where $m=0, 1, 2, \dots, (N/4)-1$ and $\theta_{m,n}=(2\pi mn)/N$.

Fig. 2 The inner structure of the DDF cell and the SDF cell.

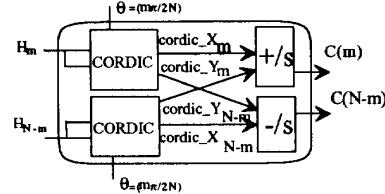


Fig. 5 The structure of the H-to-C cell.

CORDIC	NOP	$Y_0 = \text{CORDIC}(0,0)$	$-Y_0 = \text{CORDIC}(2,0)+Y_0$	$Y_1 = \text{CORDIC}(3,0)+Y_0$	$Y_1 = \text{CORDIC}(4,0)+Y_1$	$-$
DDF 2 $A_{mn} =$ $B_{mn} =$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	\dots
CORDIC 2	NOP	$Y_0 = \text{CORDIC}(0,2)$	$-Y_0 = \text{CORDIC}(2,2)+Y_0$	$Y_1 = \text{CORDIC}(3,2)+Y_0$	$Y_1 = \text{CORDIC}(4,2)+Y_1$	\dots
DDF 1 $A_{mn} =$ $B_{mn} =$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	\dots
\dots	\dots	\dots	\dots	\dots	\dots	\dots
CORDIC 3	NOP	$Y_0 = \text{CORDIC}(0,3)$	$-Y_0 = \text{CORDIC}(2,3)+Y_0$	$Y_1 = \text{CORDIC}(3,3)+Y_0$	$Y_1 = \text{CORDIC}(4,3)+Y_1$	\dots
DDF 4 $A_{mn} =$ $B_{mn} =$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	\dots
CORDIC 4	NOP	$Y_0 = \text{CORDIC}(0,4)$	$-Y_0 = \text{CORDIC}(2,4)+Y_0$	$Y_1 = \text{CORDIC}(3,4)+Y_0$	$Y_1 = \text{CORDIC}(4,4)+Y_1$	\dots
DDF 6 $A_{mn} =$ $B_{mn} =$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	\dots
\dots	\dots	\dots	\dots	\dots	\dots	\dots
CORDIC 5	NOP	$Y_0 = \text{CORDIC}(1,5)$	$-Y_0 = \text{CORDIC}(2,5)+Y_0$	$Y_1 = \text{CORDIC}(3,5)+Y_0$	$Y_1 = \text{CORDIC}(4,5)+Y_1$	\dots
DDF 7 $A_{mn} =$ $B_{mn} =$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	$(h_0+h_8)+(h_{16}+h_{24})$ $(h_0+h_8)+(h_{16}+h_{24})$	$(h_0-h_8)+(h_{16}-h_{24})$ $(h_0-h_8)+(h_{16}-h_{24})$	\dots
CORDIC 7	NOP	$Y_0 = \text{CORDIC}(1,7)$	$-Y_0 = \text{CORDIC}(2,7)+Y_0$	$Y_1 = \text{CORDIC}(3,7)+Y_0$	$Y_1 = \text{CORDIC}(4,7)+Y_1$	\dots
CONS Cell-1	NOP	\dots	\dots	$H_0H_1H_2H_3$	$H_0H_1H_2H_3$	\dots
CONS Cell-2	NOP	\dots	\dots	$H_0H_1H_2H_3$	$H_0H_1H_2H_3$	\dots
H-to-C	NOP	\dots	\dots	$C_0C_1C_2C_3$	$C_0C_1C_2C_3$	\dots

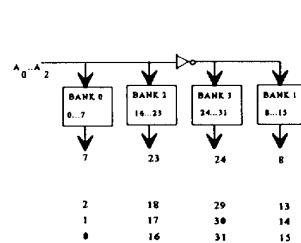
** where $\text{CORDIC}(m,n)$ consists of $\text{CORDIC}_Y(m,n) = A_{mn}\sin(2\pi mn/N) + B_{mn}\cos(2\pi mn/N)$ and $\text{CORDIC}_X(m,n) = A_{mn}\cos(2\pi mn/N) - B_{mn}\sin(2\pi mn/N)$.Fig. 7 The data flow of the proposed systolic array for the DCT with $N=32$. Double data folding are used in this design. It shows that the throughput is eight DCT transforms per clock and the latency is $5, (N/4)^2+3$.

Fig. 8 A memory addressing method

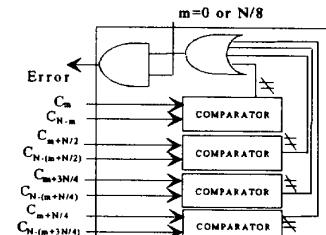


Fig. 9 The error detector and its function.