

ỨNG DỤNG THUẬT GIẢI CORDIC VÀO VIỆC TÍNH TOÁN DCT

Vũ Đình Thành ⁽¹⁾, Bùi Hữu Phú ⁽²⁾

⁽¹⁾ Trường ĐH Bách Khoa – ĐHQG-HCM, ⁽²⁾ Trung tâm kiểm soát tần số khu vực II
(Bài nhận ngày 05 tháng 07 năm 2003)

TÓM TẮT: Bài viết này sẽ đề cập đến một kỹ thuật mới trong việc tính toán biến đổi DCT (Discrete Cosine Transform) dùng thuật giải CORDIC (COordinate Rotation DIgital Computer). Nhờ thuật giải này sẽ tăng được tốc độ tính toán trên phần cứng. Các tính toán sẽ được thực hiện trên số thực dưới dạng mô tả dấu chấm động 16 bit bất chuẩn IEEE 32 bit. Mô hình phần cứng được thực hiện bằng ngôn ngữ VHDL.

I. GIỚI THIỆU:

Thuật giải CORDIC (từ viết tắt của COordinate Rotation DIgital Computer) được nhà toán học Jack E. Volder đưa ra năm 1959 dùng cho các mục đích tính toán nhanh các hàm lượng giác và các biến đổi ngược của chúng. Ưu điểm của Cordic so với các phương pháp tra bảng là tính toán nhanh mà chỉ cần lưu một ít các hệ số ban đầu.

Phép biến đổi Cosine rời rạc (DCT) được Ahmed đưa ra vào năm 1974. Kể từ đó đến nay, nó được ứng dụng rộng rãi trong nhiều phương thức mã hóa ảnh khác nhau nhờ hiệu suất gần như tối ưu của nó đối với các ảnh có độ tương quan cao giữa các điểm ảnh lân cận.

Biến đổi Cosine rời rạc được sử dụng trong chuẩn nén ảnh JPEG và định dạng phim MPEG.

II. THUẬT GIẢI CORDIC

Thuật giải CORDIC xuất phát từ các phương trình quay vector. Nếu một vector V có tọa độ là (x,y) quay một góc ϕ trong mặt phẳng xOy tạo ra một vector V' mới tọa độ là (x',y') (hình 1) thỏa mãn phương trình:

$$x' = x \cos \phi - y \sin \phi = \cos \phi [x - y \tan \phi] \quad (1)$$

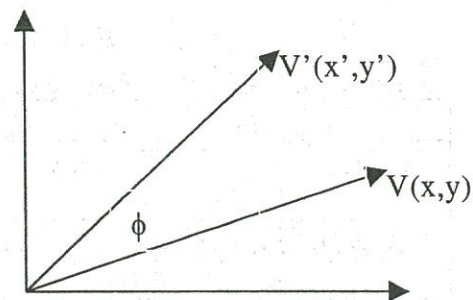
$$y' = y \cos \phi + x \sin \phi = \cos \phi [y + x \tan \phi] \quad (2)$$

Việc nhân các thành phần $\tan \phi$ có thể tránh được nếu các góc quay luôn thỏa mãn điều kiện $\tan \phi = 2^{-i}$. Trong thiết kế phần cứng, việc này đơn giản thực hiện bằng cách dịch phải i bước. Với $\phi = \arctan (2^{-i})$ thành phần $\cos(-\phi) = \cos(\phi)$ được coi là một hằng số với một số bước dịch. Việc quay này có thể được biểu diễn như sau:

$$x_{i+1} = K_i [x_i - y_i \cdot q_i \cdot 2^{-i}] \quad (3)$$

$$y_{i+1} = K_i [y_i + x_i \cdot q_i \cdot 2^{-i}] \quad (4)$$

$$\text{Trong đó : } K_i = \cos(\tan^{-1} 2^{-i}) = 1 / (1 + 2^{-2i})^{1/2} \quad d_i = +/- 1$$



Hình 1: Quay vector V một góc ϕ

Bỏ qua thành phần hằng số trong phương trình dịch, dùng một thuật giải dịch và cộng cho quay vector. Thành phần K_i có thể dùng bất cứ nơi nào trong hệ thống hay xem như là một hằng số khuếch đại. Hằng số này sấp xỉ 0.6073 khi số bước dịch đủ lớn. Vì vậy thuật giải

quay vector có một độ lợi gần bằng 1.647. Độ lợi chính xác phụ thuộc vào số bước dịch như sau:

$$A_n = \prod_n \sqrt{1 + 2^{-2i}} \tag{5}$$

Bộ quay CORDIC bình thường hoạt động ở một trong hai mode. Mode đầu gọi là mode quay, tức là quay một vector ứng với một góc nào đó. Mode thứ hai gọi là mode vector, quay một vector đến trục x.

Với mode quay, các phương trình CORDIC sẽ là:

$$x_{i+1} = x_i - y_i \cdot q_i \cdot 2^{-i} \tag{6}$$

$$y_{i+1} = y_i + x_i \cdot q_i \cdot 2^{-i} \tag{7}$$

$$z_{i+1} = z_i - q_i \cdot \tan^{-1}(2^{-i}) \tag{8}$$

Trong đó : $q_i = -1$ nếu $z_i < 0$, và $q_i = +1$ nếu $z_i \geq 0$

Và kết quả của mode quay sẽ là:

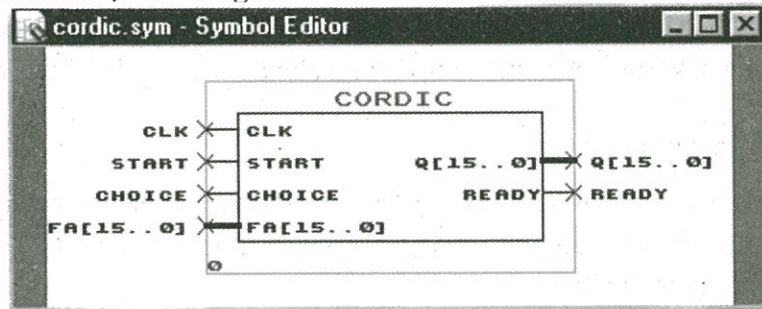
$$x_n = A_n [x_0 \cos z_0 - y_0 \sin z_0] \tag{9}$$

$$y_n = A_n [y_0 \cos z_0 + x_0 \sin z_0] \tag{10}$$

$$z_n = 0$$

Nếu cho : $x_0 = 1/A_n, y_0 = 0$ thì $x_n = \cos z_0; y_n = \sin z_0$.

Kết quả biểu diễn trên phần cứng như hình 2.



Hình 2 : Sơ đồ chân của chip CORDIC

Ý nghĩa các chân và kiểm tra tính toán :

Ngõ vào	Ngõ ra	Kiểm tra thử	Kết quả tính
Choice : 1 tính cos 0 tính sin	Ready : 1 tính xong	Choice = 1 (tính cos) Choice = 0 (tính sin)	Q = -0.503 Q = 0.862
Start : 0 chưa tính 1 cho tính	Q : kết quả tính	Start = 1	
FA : góc vào (rad)		FA = 2π/3	

III. BIẾN ĐỔI DCT

1. Các Định Nghĩa Biến Đổi DCT_1D

- FDCT_1D (DCT thuận một chiều):

$$X(k) = \sqrt{\frac{2}{N}} e(k) \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right) \text{ với } k=0,1,\dots,N-1 \tag{11}$$

- IDCT_1D (DCT ngược một chiều):

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} e(k) \cdot X(k) \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right) \text{ với } n=0,1,\dots,N-1 \tag{12}$$

trong đó:
$$e(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{nếu } k = 0 \\ 1 & \text{nếu } k \neq 0 \end{cases} \quad (13)$$

Có thể biểu diễn biến đổi DCT_1D thuận ngược dưới dạng ma trận :

$$(X_N) = [C_N] \cdot (x_N) \quad (14)$$

$$(x_N) = [C_N]^T \cdot (X_N) \quad (15)$$

trong đó X_N và x_N là các vector cột của các ngõ ra $X(k)$ và ngõ vào $x(n)$

C_N là ma trận $N \times N$ phần tử các hệ số lượng giác, với $N=8$, C_N xác định như sau :

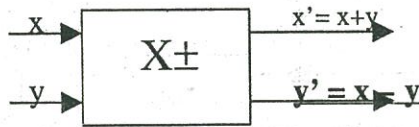
$$C_N = \begin{pmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & -C_7 & -C_5 & -C_3 & -C_1 \\ C_2 & C_6 & -C_6 & -C_2 & -C_2 & -C_6 & C_6 & C_2 \\ C_3 & -C_7 & -C_1 & -C_5 & C_5 & C_1 & C_7 & -C_3 \\ C_4 & -C_4 & -C_4 & C_4 & C_4 & -C_4 & -C_4 & C_4 \\ C_5 & -C_1 & C_7 & C_3 & -C_3 & -C_7 & C_1 & -C_5 \\ C_6 & -C_2 & C_2 & -C_6 & -C_6 & C_2 & -C_2 & C_6 \\ C_7 & -C_5 & C_3 & -C_1 & C_1 & -C_3 & C_5 & -C_7 \end{pmatrix} \quad (16)$$

Trong đó kí hiệu : $C_m = \cos(m\pi/16)$ (17)

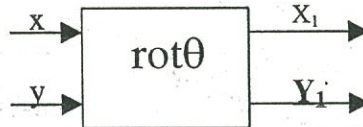
Chú ý : trong công thức dạng ma trận, ta bỏ qua hệ số hằng $\sqrt{\frac{2}{N}}$

2. Cấu Trúc VLSI Cho DCT Một Chiều :

Trong cấu trúc VLSI thực hiện phép tính DCT, hai phần tử căn bản được thiết kế là khối X_{\pm} được biểu diễn ở hình 3 và khối xoay góc θ , là $rot\theta$ được biểu diễn trong hình 4.



Hình 3 : Định nghĩa khối X_{\pm}



Hình 4 : Định nghĩa khối $rot\theta$

Trong đó : X_1 và Y_1 được xác định theo phương trình sau :

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (18)$$

và $\theta_i = (i\pi/16)$; $C_i = \cos(\theta_i)$

Nhờ các phương trình (14), (15), (16), (18), ta xây dựng cấu trúc VLSI của biến đổi FDCT_1D như ở hình 5.

Trong đó $x(i)$ ($i = 0, \dots, 7$) là các ngõ vào, $X(j)$ ($j = 0, \dots, 7$) là các ngõ ra của FDCT_1D.

M_i, P_i, M_{ij}, P_{ij} là các ký hiệu ngõ vào và ra của các khối, C_4 là hệ số nhân $\cos(\theta_4)$.

3. Biến Đổi DCT Hai Chiều (DCT_2D)

Biến đổi DCT_2D được xác định theo các phương trình sau:

- Biến đổi thuận (FDCT_2D)

$$X(i, j) = \sqrt{\frac{1}{2N}} e(i).e(j). \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} z(x, y). \cos\left(\frac{(2x+1).i\pi}{2N}\right). \cos\left(\frac{(2y+1).j\pi}{2N}\right) \quad (19)$$

- Biến đổi ngược (IDCT_2D)

$$z(x, y) = \sqrt{\frac{1}{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} e(i).e(j).X(i, j). \cos\left(\frac{(2x+1).i\pi}{2N}\right). \cos\left(\frac{(2y+1).j\pi}{2N}\right) \quad (20)$$

-Dạng ma trận:

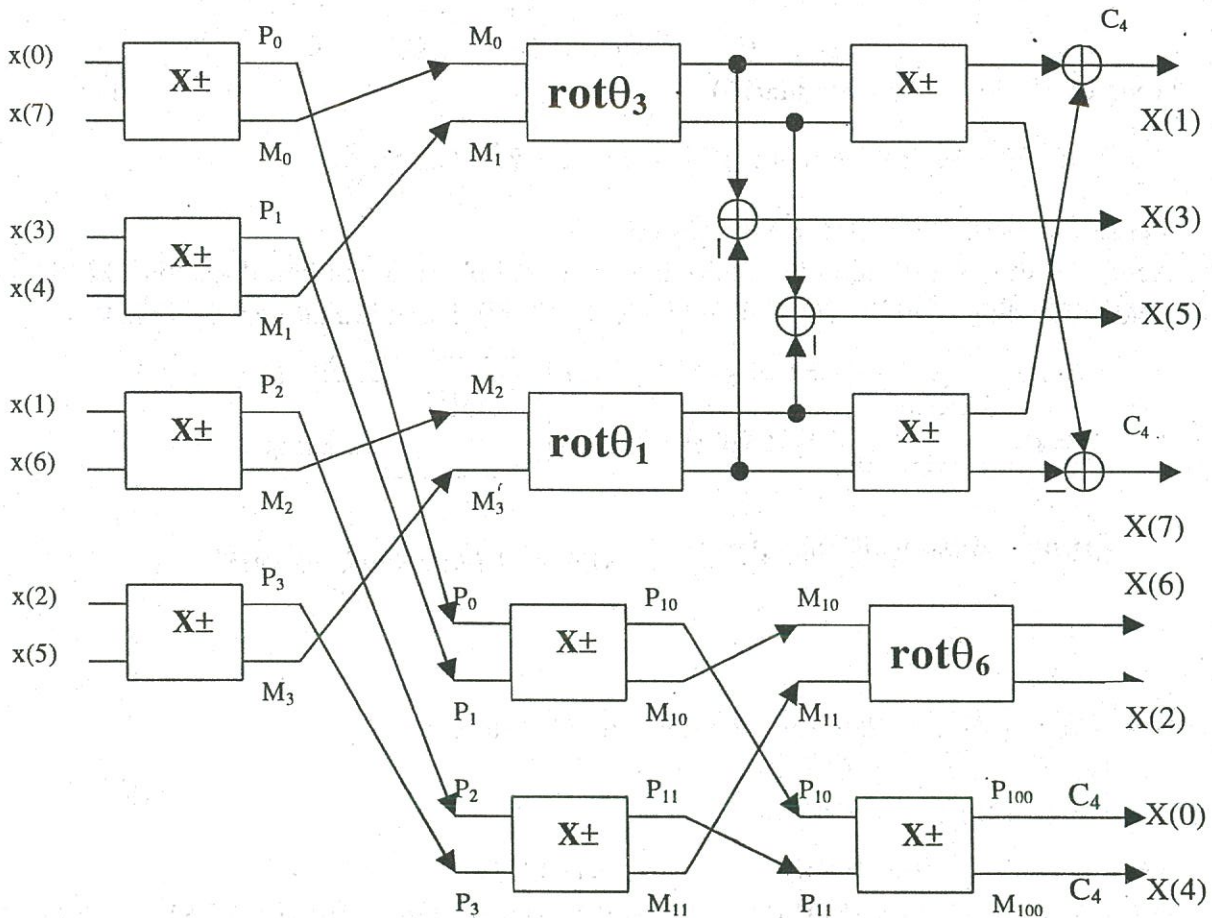
$$\mathbf{X}_{NN} = \mathbf{C}_N . \mathbf{Z}_{NN} . [\mathbf{C}_N]^T \quad (21)$$

$$\mathbf{Z}_{NN} = [\mathbf{C}_N]^T . \mathbf{X}_{NN} . \mathbf{C}_N \quad (22)$$

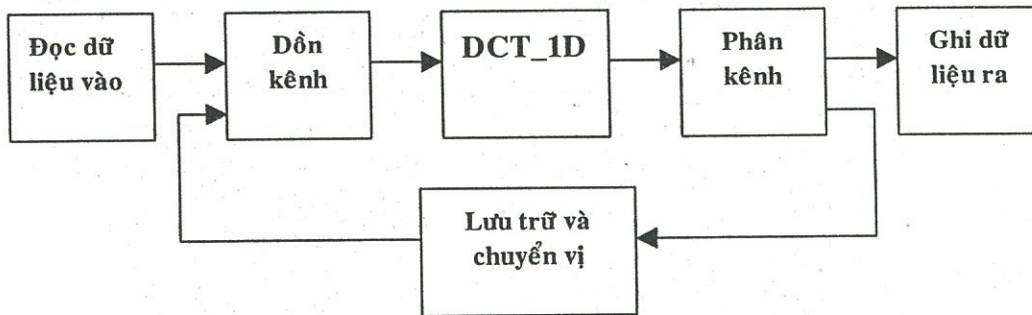
Trong đó \mathbf{Z}_{NN} và \mathbf{X}_{NN} là các ma trận vào và ra tương ứng.

Ta hoàn toàn có thể phân tích biến đổi DCT_2D thành các biến đổi một chiều trên các dòng – cột, sơ đồ khối cho DCT_2D như trong hình 6.

Các biến đổi IDCT_1D và IDCT_2D cũng làm tương tự, đối với IDCT_2D, dựa trên các biến đổi ngược một chiều IDCT_1D, sơ đồ khối cho IDCT_2D tương tự hình 6, chỉ thay khối DCT_1D bằng IDCT_1D.

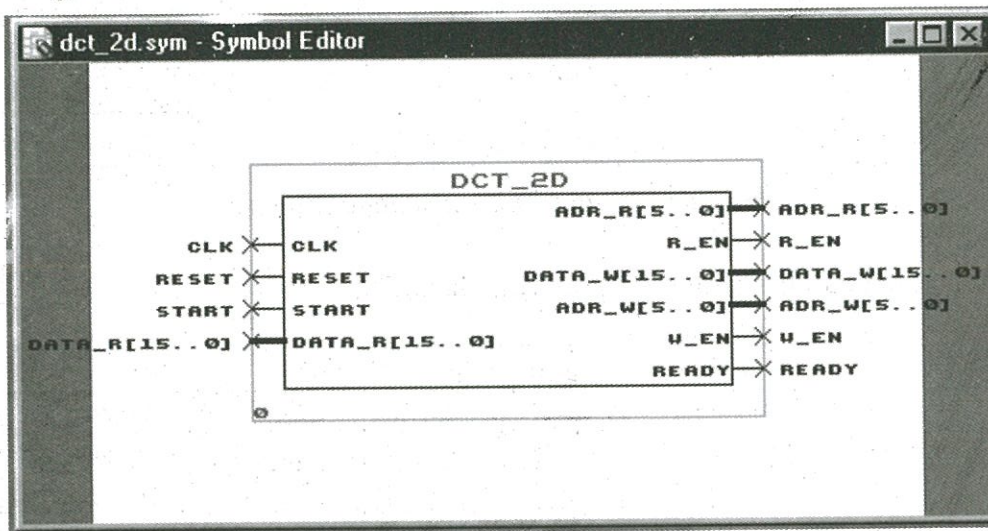


Hình 5 : Cấu trúc VLSI cho DCT_1D với 13 phép nhân



Hình 6 : Sơ đồ khối phần cứng biến đổi DCT_2D

Việc thực hiện mô hình phần cứng cho DCT_2D bằng ngôn ngữ VHDL có mô hình kết quả trong Hình 7.



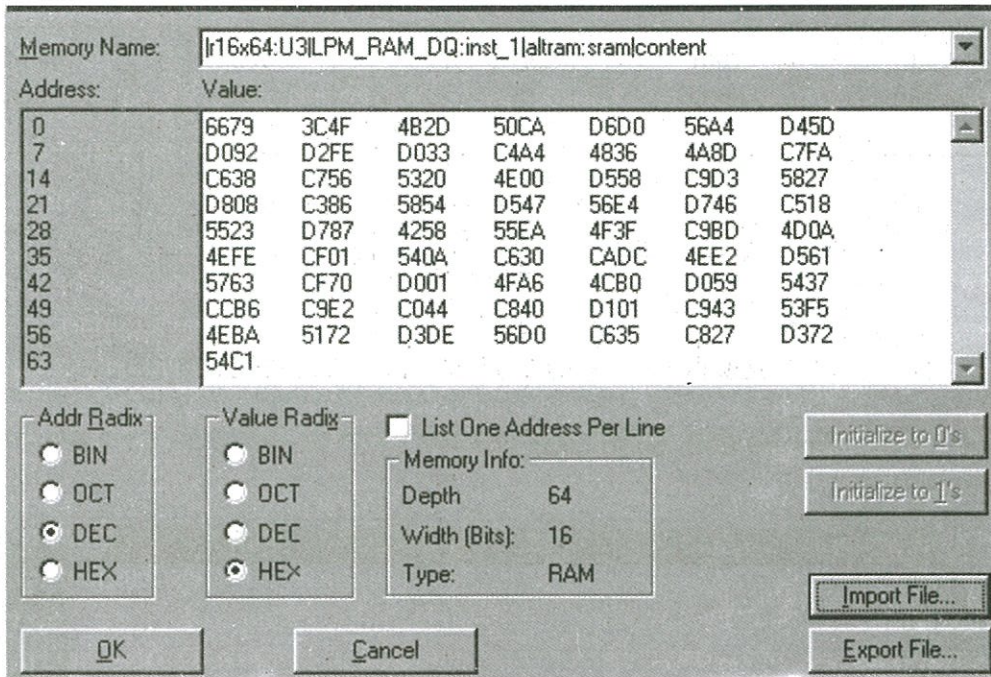
Hình 7 : Kết quả thực hiện DCT_2D

IV. TÍNH TOÁN CỤ THỂ

Khối dữ liệu đưa vào có các giá trị dạng thập phân cho trong bảng sau:

Dữ liệu vào							
50	43	53	50	56	43	56	61
50	48	58	50	56	58	56	58
53	45	45	40	45	61	50	61
45	58	50	58	56	50	50	61
81	48	38	-21	45	53	50	61
40	45	68	73	56	43	50	58
48	40	89	89	61	40	58	73
45	40	48	66	7	45	48	53

Kết quả biến đổi DCT_2D của dữ liệu bằng chip DCT_2D như trong hình 8



Hình 8 : Kết quả DCT_2D của dữ liệu trên

Kết quả tính toán biến đổi DCT_2D dạng decimal.

1657	-55.93	57	-84.437	28.984	27.531	67.437	22.906
1.077	-33.593	24	110.25	-11.476	-86.062	-18.843	43.562
14.351	-4.64	-85	-116.375	20.156	118.187	-11.765	-63.437
38.31	8.42	-11.648	-5.093	27.968	-29.75	-2.132	109
-109	13.101	132.875	82.187	-28.015	-32.031	-8.5	-6.207
106.25	-7.976	-129	-120.437	64.625	30.593	-40.031	-8.304
-69.81	-6.218	-3.761	3.171	-6.187	18.75	-10.523	-59.56
-36.562	-7.335	138.5	94.625	-13.718	-34.781	63.656	76.062

1655.7	-56.5	51.8	-84.5	28.8	27.3	67.1	26.7
1.12	-33.63	23.92	110.29	-11.37	-86.02	-18.90	43.51
14.4	-4.6	-85.5	-116.4	20.2	118.3	-11.8	-62.9
38.3	8.4	-11.6	-5.1	27.9	-29.8	-2.1	109
-109	13.1	132.9	82.3	-28.0	-32.0	-8.5	-6.2
106.3	-8.0	-129.1	-120.5	64.7	30.6	-40.1	-8.3
-69.8	-6.2	-3.8	3.2	-6.2	18.8	-10.5	-59.6
-36.6	-7.2	138.5	94.7	-13.7	-34.8	63.7	76.1

Lưu ý rằng kết quả biến đổi DCT_2D trên đây chưa kể đến hệ số $\sqrt{\frac{1}{2N}}$

V. KẾT LUẬN:

Mô hình phần cứng thực hiện cho thuật giải Cordic trên đây áp dụng được cho mọi góc (lớn tùy ý trong phạm vi có thể mô tả của số floating point 16 bit). Hơn nữa việc tính toán thực hiện trên số thực (real), do đó mô hình có tính tổng quát cao, tuy nhiên sẽ chiếm bộ nhớ lớn hơn. Các phép tính sin, cos sai số ở số lẻ thứ ba, độ chính xác có thể được cải tiến hơn nữa, nếu tăng số bit mô tả số floating point.

Mô hình xây dựng tính toán DCT trong hình 5 và 6 được thực hiện trên số thực 16 bit, có thể áp dụng cho bất kỳ ứng dụng nào đòi hỏi giá trị của các số liệu là lớn hay độ chính xác cao. Tuy nhiên, khi thực hiện trên FPGA số logic cells đòi hỏi lớn.

Với việc ứng dụng thuật giải Cordic vào việc tính toán DCT có thể sẽ được dùng nhiều trong các công nghệ nén và giải nén ảnh số trong tương lai.

APPLICATION OF THE CORDIC ALGORITHM INTO THE CALCULATION OF DISCRETE COSINE TRANSFORM COEFFICIENTS

Vu Dinh Thanh, Bui Huu Phu

***ABSTRACT:** This paper presents a new approach in the calculation of DCT (Discrete Cosine Transform) coefficients by using CORDIC algorithm (COordinate Rotation DIgital Computer). Hardware processing will be enhanced by applying this algorithm. The computation will be carried out on the real number indicated under 16-bit floating point following the similar presentation as in IEEE 32-bit. The results are simulated in the VHDL environment.*

TÀI LIỆU THAM KHẢO

- [1] Ray Andraka, A Survey Of Cordic Algorithms For Fpga Based Computers, (*.pdf file).
- Tanya Vladimirova and Hans Tiggeler, Fpga Implementation Of Sine And Cosine Generators
- [2] Using The Cordic Algorithm, (*.pdf file).
- [3] <http://www.andraka.com/cordic.htm>.
- [4] Altera Corporation , “Max+plusII – programmable Logic Development System”, 1994.
- [5] Lương Mạnh Bá và Nguyễn Thanh Thủy, Nhập Môn Xử Lý Ảnh Số, Nhà xuất bản khoa học kỹ thuật (1999).