Efficient and High Accuracy 2-D DCT and IDCT Architecture for Image Compression Based on Improved CORDIC

WU Hao[1, 2], SONG Helun[2], LIU Nan[2], DING Peng[2], WU Fei[2], LI Zhenyao[2], WANG Zhengguang[1,2], JI Yu[1,2], RU Zhanqiang[2]

(1.*School of Nano-Tech and Nano-Bionics*, *University of Science and Technology of China*, *Hefei*, 230026, *China*; 2. *Suzhou Institute of Nano-Tech and Nano-Bionics*, *Chinese Academy of Sciences*, *Suzhou* 215123, *China*)

Abstract: Discrete Cosine Transform (DCT) is the most widely used technique in image and video compression. In this paper, the structure of DCT and Inverse DCT (IDCT) algorithm is split in the form of COordinate Rotation DIgital Computer (CORDIC) rotation matrix. The two-dimensional (2-D) 8×8 DCT/IDCT units based on the improved rotation CORDIC algorithm is proposed. The shift and addition operations of the CORDIC algorithm are used to replace the cosine multiplication operations in the algorithm. The design does not contain any multiplier unit, which reduces the complexity of the hardware unit. The row-column transform unit composed of register arrays connects two 1-D 8-point DCT units to complete the calculation of 2-D 8×8 DCT. The pipeline latency of proposed architecture is 28 clock cycles. The proposed efficient two-dimensional DCT architecture has been synthesized on the Xilinx's Kintex-7 FPGA. The resource utilization is 17.36 % for Slice LUTs, 3.49 % for Slice Registers, and the maximum operating frequency is 172 MHz. It takes only 0.161μs to complete a process of block of 8×8 samples. A frame of image is processed by the designed DCT unit and then reconstructed by the IDCT unit to verify the function. The Peak Signal to Noise Ratio (PSNR) can reach 51.99 dB.

Keywords: DCT, IDCT, Image and Video Compression, CORDIC, FPGA

1 Introduction

Discrete cosine transform is a kind of frequency domain transform, which has excellent energy concentration characteristics[1]. It is widely used in image and video compression[2-3], such as JPEG[4], H.263[5], MPEG[6] and HEVC[7]. However, a large amount of data calculation is involved in DCT calculation. The complexity of 1D-DCT algorithm is $O(N^2)$, while the image is a two-dimensional signal, and the complexity of 2D-DCT algorithm reaches $O(N^4)$. Therefore, scholars at home and abroad continue to study and improve the acceleration of DCT.

The multiplication of cosine function is mainly need to be completed in DCT. The main ideas to accelarate this calculation are as follows: reducing the number of multipliers, realizing distributed computing, realizing the architecture without multiplier through shift and addition operations or using CORDIC algorithm.

A fast 8-point DCT algorithm, which classifies

the calculation according to the characteristics of trigonometric functions was proposed[8]. This algorithm only needs 11 multiplications and 29 additions to complete the DCT, which reduces the number of multipliers. There are also some related studies about distributed computing methods. In 2013, two high-speed FPGA architectures for 2D-DCT calculation was proposed by Kitsos et al.[9] Distributed computing is completed through ROM and adders. ROM contains constant coefficients and reduces multipliers in DCT algorithms.

The multiplier structure is complex and occupies a large amount of hardware resources. Therefore, a considerable solution is based on multiplier-free design, which replaces the multiplier by shifting and adding operations. In 2000, Tran et al.[10] proposed a fast biorthogonal block transform called binDCT, which used only shift and addition operations to achieve a new 8×8 DCT architecture, obtaining a coding gain range of 8.77-8.82 dB, and only 14 shifts and 31 additions were required per 8 input samples. In 2011, Cintra et al.[11] proposed an 8-point DCT approximation algorithm based on matrix polar decomposition. The transform matrix in DCT only contains 0 and 1, without multiplication and shift operations.

Researchers classify the cosine functions in DCT transform and find that they can be decomposed in the form of iterative matrix in CORDIC algorithm, and the design without multiplier can also be realized.

In 2002, Yu et al.[12] proposed the scalable DCT algorithm by using the CORDIC algorithm and the indirect calculation method of vector rotation and other operations separation. The algorithm does not require scale factor compensation, and the vector rotation is placed at the last level of the DCT unit, but a DCT unit contains 6 CORDIC modules, which takes up more hardware resources. In 2006, Sung et al.[13] designed an 8-point DCT/IDCT unit with 5 CORDIC modules, and used RAM as the buffer block of intermediate data to design a small area and low power consumption 2-D 8×8 DCT processor. However, the processor has a calculation latency of 64 clock cycles. In 2019, a unified DCT/IDCT architecture based on an improved enhanced adaptive coding CORDIC algorithm was proposed by Zhang Jianfeng et al.[14], and the scale factor approximation of radix-2 was optimized to obtain better power consumption and PSNR.

A certain quantization coefficient is used to mask the DCT block to achieve compression, and label bits can also be embedded in the image area discarded by the quantization coefficient processing to design the digital watermarking of the image[15].

For the solution of 2-D $N\times N$ DCT, the separability of DCT structure is usually used to calculate two 1-D N-point DCT in two different stages by row and column decomposition technique. The results of first stage are transformed by rows and columns to complete the 1-D N-point DCT of N-column's samples in second stage. Meher et al.[16], Srivastava et al.[17] and related research[18] all implement the DCT architecture according to this method.

2 Principles of the CORDIC and DCT

2.1 Conventional CORDIC Algorithm

The CORDIC algorithm was proposed by Volder et al.[19] in 1959 to solve the calculation of trigonometric functions in aviation navigation systems. The algorithm contains only shift and addition operations and is an efficient hardware algorithm. It works in two modes: rotation and vector, which can be applied to circular system, linear system and hyperbolic system to solve various transcendental functions.

Since DCT and IDCT algorithms are mainly composed of cosine functions, the rotation mode in circular system of the traditional CORDIC algorithm can be used. The coordinate rotation under the circular system is shown in Fig.1.

Assuming that there is a point $P(x_0,\ y_0)$ rotates angle of θ to get $P'(x_N,\ y_N)$, therefore the relation between the two points is shown in Equation (1).

$$\begin{cases} x_N = x_0 \cdot \cos\theta - y_0 \cdot \sin\theta \\ y_N = x_0 \cdot \sin\theta + y_0 \cdot \cos\theta \end{cases} \tag{1}$$

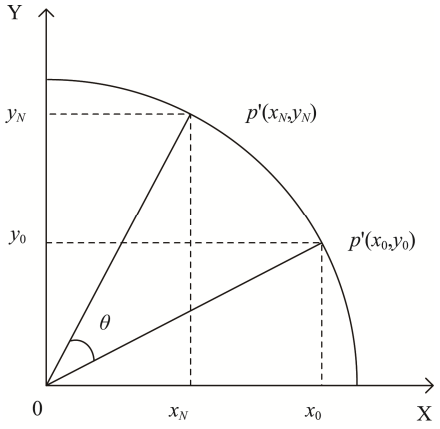

Fig.1 Rotation in Circular System

Adopting the idea of iterative decomposition, the rotation angle θ is decomposed into several angles as θ_n, that is, each iteration is to rotate a small angle, which is also called the micro-rotation angle. Therefore, according to Equation (1), the iterative relationship of each step is further derived.

$$\begin{cases} x_{n+1} = \cos\theta_n \cdot (x_n - y_n \cdot \tan\theta_n) \\ y_{n+1} = \cos\theta_n \cdot (x_n \cdot \tan\theta_n + y_n) \end{cases} \tag{2}$$

Where, θ_n has the characteristics shown in Equation (3).

$$\begin{cases} \theta_n = \tan^{-1}(\delta_n \cdot 2^{-n}) \\ \theta = \sum_{i=0}^{\infty} \theta_n \end{cases} \tag{3}$$

Where, n: 0, 1, 2, …, N-1, is a iteration number; $\delta_n \in \{-1,1\}$.

After N times of rotation, the relationship between the starting point and the corresponding point after the rotation angl

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \prod_{n=0}^{N-1} \cos\theta_n \cdot \begin{bmatrix} 1 & -\delta_n \cdot 2^{-n} \\ \delta_n \cdot 2^{-n} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{4}$$

Where, $\prod_{n=0}^{N-1} \cos\theta_n$ (n>12) will converge to a constant, let the constant be

$$K = \prod_{n=0}^{N-1} \cos\theta_n = \prod_{n=0}^{N-1} \left(\frac{1}{\sqrt{1+2^{-2n}}}\right) \approx 0.6072529 \tag{5}$$

Introducing the rotation residual angle z_n as

$$z_n = z_0 - \delta_n \cdot \arctan(2^{-n}) \tag{6}$$

Where, after N times of rotation the initial value of the rotation residual angle z_0 reduces from θ to 0. Then, when z_n is positive, the δ_n is 1. Otherwise, the δ_n is -1. Finally, the iterative result is shown in Equation (7).

$$\begin{cases} x_N = \frac{1}{K} \cdot (x_0 \cdot \cos z_0 - y_0 \cdot \sin z_0) \\ y_N = \frac{1}{K} \cdot (x_0 \cdot \sin z_0 + y_0 \cdot \cos z_0) \\ z_N = 0 \end{cases} \tag{7}$$

2.2 1-D DCT and IDCT Algorithm

1-D DCT transforms N-point sample $x(n)$ from time domain to frequency domain signal $y(k)$, defined as

$$y(k) = \sqrt{\frac{2}{N}} \cdot c(k) \cdot \sum_{n=0}^{N-1} x(n) \cdot \cos\left[\frac{(2n+1)k\pi}{2N}\right] \tag{8}$$

Where, k = 0, 1, 2, … , N-1, c(k) is the scaling factor. When k is 0, c(k) is $1/\sqrt{2}$. Otherwise, c(k) is 1. The complexity of the direct calculation algorithm is O(N^2), and the cosine value needs to be stored, which is restricted by the multiplier unit and the memory.

1D-IDCT is the inverse transform of 1D-DCT. The N-point frequency domain signal $y(k)$ (k = 0, 1, 2,…, N-1) is converted into the time domain signal $x(n)$ (n = 0, 1, 2, …, N-1). The Equation is

$$x(n) = \sqrt{\frac{2}{N}} \cdot \sum_{k=0}^{N-1} c(k) \cdot y(k) \cdot \cos\left[\frac{(2n+1)k\pi}{2N}\right] \tag{9}$$

Where, c(k) is the scaling factor, which is consistent with the definition in Equation (8).

2.3 2-D DCT Algorithm

Since both image and video signals are stored in two-dimensional form, for 2-D $N\times N$ sample $x(m,n)$ (m, n = 0, 1, 2, … , N-1), it is converted from time domain to frequency domain $y(p,q)$ (p, q = 0, 1, 2, …, N-1), and the Equation is

$$\begin{aligned} y(p,q) &= \frac{2c(p)c(q)}{\sqrt{N \times N}} \\ &\cdot \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) \cdot \cos\left[\frac{(2m+1)k\pi}{2N}\right] \cdot \cos\left[\frac{(2n+1)k\pi}{2N}\right] \end{aligned} \tag{10}$$

Where, the scaling factors c(p) and c(q) are

consistent with c(k) in 1D-DCT. When p or q is 0, the scaling factor is 0. Otherwise, the scaling factor is 1.

According to Equation (10), the computational complexity of the 2-D $N\times N$ DCT algorithm is O(N^4). Therefore, the direct hardware implementation of the 2D-DCT is usually not considered, and the separability of the 2D-DCT algorithm is used: Firstly, complete the calculation of N row 1-D N-point DCT in block of $N\times N$ samples, and then the calculated results are transformed by row-column transform unit to complete the DCT calculation of N column 1-D N-point. Therefore, this method can be used for the hardware implementation of 2D-DCT.

3 Proposed Architecture

3.1 Improved CORDIC Architecture

The conventional CORDIC algorithm requires at least 16 iterations to complete a certain precision calculation, which is unbearable in scenarios with high real-time requirements. It is necessary to reduce the number of iterations in the algorithm and improve the calculation speed.

In the rotation mode, since each iteration needs to judge the positive and negative of the remaining angle to determine the direction of rotation, so the angle of each two-step iteration can be coded. The main method is to judge the positive and negative of the current remaining angle and the result of remaining angle adding or substracting the current iterative micro-rotation angle. Therefore, it is only necessary to judge the interval of the remaining angle, which can be coded according to the angle, realize two-step iteration and reduce the total number of iterations.

As shown in Table 1, in the first iteration, the remaining rotation angle is the initial value z_0. First, δ_0 can be obtained by judging the positive and negative of z_0, and then δ_1 can be obtained by adjusting the positive or negative of z_0 adding or subtracting $\theta_0=\tan^{-1}(2^0)=45°$, and then x_2, y_2 and z_2 can be calculated directly.

Table 1 Remaining Angle Coding in First Iteration

θ(°)	δ_0	δ_0
[-180,-45)	-1	-1
[-45,0)	-1	1
[0.45)	1	-1
[45.180)	1	1

Using the method of remaining angle coding, δ_{2n}, $\delta_{2n+1}(n = 0, 1, 2, \cdots, n)$ can be directly calculated in the n^{th} iteration, so that the overall number of iterations reduces by half and the calculation speed is improved.

In this way, only 8 times of iterations are needed to complete the calculation with the same accuracy by 16 times of iterations of conventional CORDIC algorithm, and the scaling operation of the calculation results is completed in the last stage.

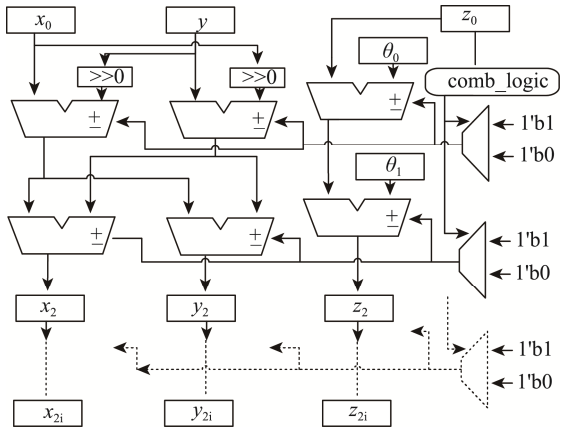

Fig.2 Architecture of Improved Rotation CORDIC

The Verilog hardware description language is used to complete the register transfer level (RTL) implementation of the improved CORDIC algorithm. The SystemVerilog verification platform is built and the function model of the C language is introduced through the 'DPI-C' interface. The random data generator is used to generate data, which is input into the reference model and the design under test, and the calculated results are compared. Compile Testbench and RTL design through VCS software. The sine and cosine functions are calculated by one million random data tests. The result of reference model, improved

CORDIC algorithm, the error and the mean error after all tests completed are shown in Fig.3.

```
-------------------------------------------------------
Num: 999997    Name                    Value
-------------------------------------------------------
               ref_model_cos           -0.9435366339
               cordic_cos              -0.9433898926
               cordic_cos_error        0.0001467414
               ref_model_sin           0.3312682001
               cordic_sin              0.3312072754
               cordic_sin_error        -0.0000609247
-------------------------------------------------------
Num: 999998    Name                    Value
-------------------------------------------------------
               ref_model_cos           -0.5647010225
               cordic_cos              -0.5646362305
               cordic_cos_error        0.0000647920
               ref_model_sin           -0.8252955563
               cordic_sin              -0.8251190186
               cordic_sin_error        0.0001765377
-------------------------------------------------------
Num: 999999    Name                    Value
-------------------------------------------------------
               ref_model_cos           -0.9860011954
               cordic_cos              -0.9858093262
               cordic_cos_error        0.0001918693
               ref_model_sin           0.1667382457
               cordic_sin              0.1666870117
               cordic_sin_error        -0.0000512340
-------------------------------------------------------
----------After 1000000 Tests----------------------------
-------------------------------------------------------
               cos_mean_error          -0.0000013436
               sin_mean_error          -0.0000004274
-------------------------------------------------------
```

Fig.3 Error of Sine and Cosine Functions of Improved CORDIC

3.2 Improved CORDIC Based DCT Architecture

Aiming at the problems of excessive calculation and large storage space of DCT, a 1-D 8-point DCT algorithm based on improved CORDIC algorithm is proposed. The 1-D 8-point DCT is classified and resolved by using the sine and cosine iterative matrix ignoring the scaling factor *K* in Equation (7), which is expressed by matrix multiplication.

$$\begin{bmatrix} y(0) \\ y(4) \end{bmatrix} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x(0)+x(3)+x(4)+x(7) \\ x(1)+x(2)+x(5)+x(6) \end{bmatrix}$$
$$\begin{bmatrix} y(1) \\ y(7) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(7) & c(1) \\ -c(1) & c(7) \end{bmatrix} \cdot \begin{bmatrix} x(3)-x(4) \\ x(0)-x(7) \end{bmatrix} + \frac{1}{2} \cdot \begin{bmatrix} c(3) & c(5) \\ -c(5) & c(3) \end{bmatrix} \cdot \begin{bmatrix} x(1)-x(6) \\ x(2)-x(5) \end{bmatrix}$$
$$\begin{bmatrix} y(3) \\ y(5) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(3) & -c(5) \\ c(5) & c(3) \end{bmatrix} \cdot \begin{bmatrix} x(0)-x(7) \\ x(3)-x(4) \end{bmatrix} - \frac{1}{2} \cdot \begin{bmatrix} c(1) & c(7) \\ -c(7) & c(1) \end{bmatrix} \cdot \begin{bmatrix} x(2)-x(5) \\ x(1)-x(6) \end{bmatrix}$$
$$\begin{bmatrix} y(6) \\ y(2) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(6) & -c(2) \\ c(2) & c(6) \end{bmatrix} \cdot \begin{bmatrix} x(0)+x(7)-x(3)-x(4) \\ x(1)+x(6)-x(2)-x(5) \end{bmatrix} \tag{11}$$

Where, $c(n) = \cos(\frac{n\pi}{16})$. The cosine transform matrix is transformed by law of cosine according to the format of CORDIC's iterative matrix, then the Equation (11) can be transformed into Equation (12).

$$\begin{cases} \begin{bmatrix} y(0) \\ y(4) \end{bmatrix} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x(0)+x(3)+x(4)+x(7) \\ x(1)+x(2)+x(5)+x(6) \end{bmatrix} \\ \begin{bmatrix} y(1) \\ y(7) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(7) & s(7) \\ -s(7) & c(7) \end{bmatrix} \cdot \begin{bmatrix} x(3)-x(4) \\ x(0)-x(7) \end{bmatrix} + \frac{1}{2} \cdot \begin{bmatrix} c(3) & s(3) \\ -s(3) & c(3) \end{bmatrix} \cdot \begin{bmatrix} x(1)-x(6) \\ x(2)-x(5) \end{bmatrix} \\ \begin{bmatrix} y(3) \\ y(5) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(3) & -s(3) \\ s(3) & c(3) \end{bmatrix} \cdot \begin{bmatrix} x(0)-x(7) \\ x(3)-x(4) \end{bmatrix} - \frac{1}{2} \cdot \begin{bmatrix} c(1) & s(1) \\ -s(1) & c(1) \end{bmatrix} \cdot \begin{bmatrix} x(2)-x(5) \\ x(1)-x(6) \end{bmatrix} \\ \begin{bmatrix} y(6) \\ y(2) \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(6) & -s(6) \\ s(6) & c(6) \end{bmatrix} \cdot \begin{bmatrix} x(0)+x(7)-x(3)-x(4) \\ x(1)+x(6)-x(2)-x(5) \end{bmatrix} \end{cases} \tag{12}$$

Where, $s(n) = \sin(\frac{n\pi}{16})$.

The corresponding angle and the initial value of the X and Y paths are input into the CORDIC module. Through simple shift and addition operations, the trigonometric function matrix multiplication operation in the formula is replaced to overcome the large consumption of hardware resources by the multiplication unit and the trigonometric function look up table.

The paths of y(0) and y(4) do not contain CORDIC computing units in the Equation (12), but they need to be multiplied by $1/\sqrt{8}$. Here, the coefficient equivalent conversion in hardware implementation requires only several adders and shifters. The coefficient, expressed as 0.35355335 (decimal), then the decimal is firstly converted into 0.010110101 (binary). Therefore, move the decimal point of the coefficient to the right by 9 bits, that is, multiply by 2^9 in decimal, the binary is represented as 10110101, and then divide by 2^9. The shift operation is represented in Equation (13).

$$result = (a << 7 + a << 5 + a << 4 + a << 2 + a) >> 9 \tag{13}$$

When solving other results other than y(0) and y(4), the paths contain CORDIC module, and the input angles are $\frac{6\pi}{16}$, $\frac{-7\pi}{16}$, $\frac{-3\pi}{16}$, $\frac{3\pi}{16}$, $\frac{-\pi}{16}$, respectively. The final coefficient 1/2 of the path can be realized by right shift in binary.

According to the Equation (12), the data preparation need to be completed before the coefficient transform and CORDIC calculation, such as

$$\begin{cases} s07=x(0)+x(7) \\ s16=x(1)+x(6) \end{cases}, \begin{cases} s25=x(2)+x(5) \\ s34=x(3)+x(4) \end{cases}$$
$$\begin{cases} s0\_7=x(0)-x(7) \\ s1\_6=x(1)-x(6) \end{cases}, \begin{cases} s2\_5=x(2)-x(5) \\ s3\_4=x(3)-x(4) \end{cases} \quad (13)$$

and

$$\begin{cases} s0734=s07+s34 \\ s1625=s16+s25 \end{cases}, \begin{cases} s07\_34=s07-s34 \\ s16\_25=s16-s25 \end{cases}$$
$$\begin{cases} s07341625=s0734+s1625 \\ s0734\_1625=s0734-s1625 \end{cases} \quad (14)$$

In the process of data preparation, 14 adders are used. After the CORDIC module completes the multiplication of sine and cosine, 4 adders and 6 shifters are needed. The architecture of 1-D 8-point DCT based on CORDIC algorithm is shown in Fig.4.

Connecting two 1D-DCT modules through the row-column transform unit to compose the architecture of 2D-DCT is the main idea of the current 2D-DCT implementation. In this paper, the shift register array is used to complete the row and column transform of data, which only needs 8 clock cycles. Compared with SRAM storage, the clock cycle occupied by data serial readout is eliminated. Fig.5 shows the 2-D 8×8 DCT architecture designed in this paper.

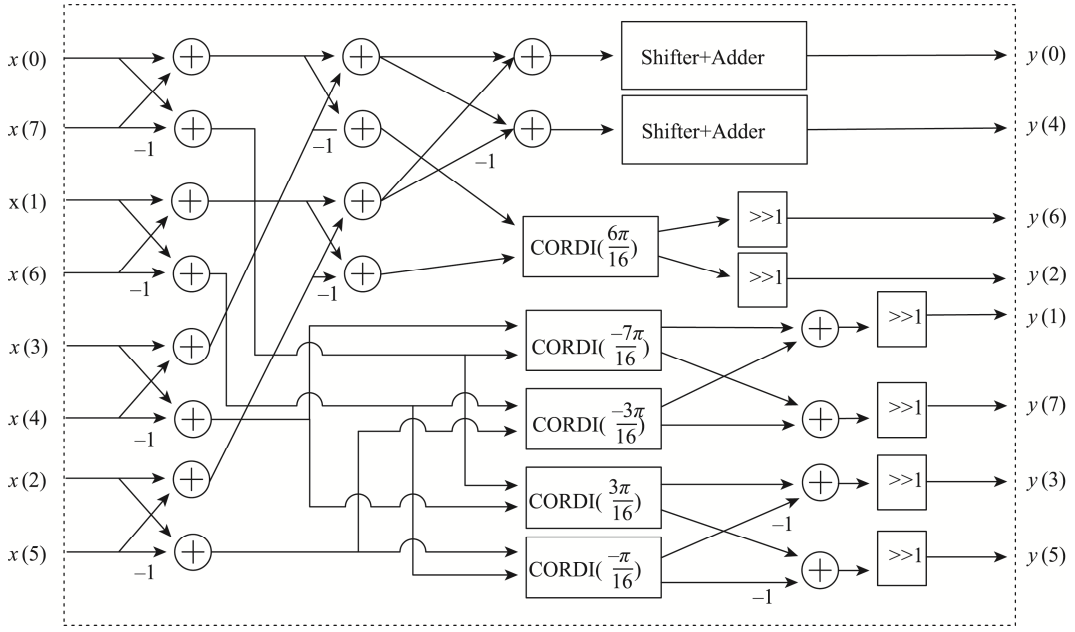

Fig.4 1-D 8-point DCT Based on CORDIC Algorithm

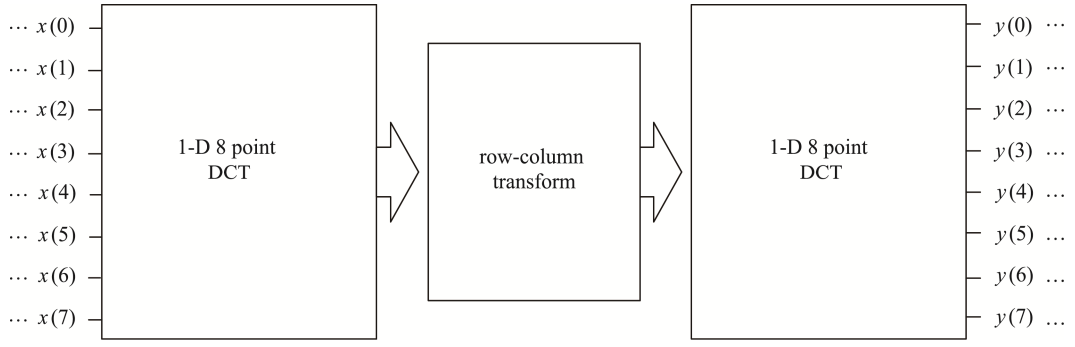

Fig.5 2-D 8×8 DCT Architecture

3.3 Improved CORDIC Based IDCT Architecture

Since IDCT is an inverse transform of DCT, the 2D-IDCT is also separable, and can be decomposed into two 1D-IDCT operations through row-column transform. According to the principle of 1D-IDCT and the implementation of 2D-DCT, the Equation is

$$\begin{cases} \begin{bmatrix} b1 \\ b0 \end{bmatrix} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} y(0) \\ y(4) \end{bmatrix} \\ \begin{bmatrix} b3 \\ b2 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(2) & -s(2) \\ s(2) & c(2) \end{bmatrix} \cdot \begin{bmatrix} y(6) \\ y(2) \end{bmatrix} \\ \begin{bmatrix} b5 \\ b4 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c(1) & -s(1) \\ s(1) & c(1) \end{bmatrix} \cdot \begin{bmatrix} y(7) \\ y(1) \end{bmatrix} \\ \begin{bmatrix} b7 \\ b6 \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} c1 & -s1 \\ s1 & c1 \end{bmatrix} \cdot \begin{bmatrix} y(3) \\ y(5) \end{bmatrix} \\ \begin{bmatrix} b9 \\ b8 \end{bmatrix} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} c(1) & -s(1) \\ s(1) & c(1) \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} y(7) \\ y(1) \end{bmatrix} \\ \begin{bmatrix} b11 \\ b10 \end{bmatrix} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} c(1) & -s(1) \\ s(1) & c(1) \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} y(3) \\ y(5) \end{bmatrix} \end{cases} \tag{15}$$

According to the Equation (9), the results of IDCT are obtained by Equation (16).

$$\begin{cases} x(0)=(b1+b2)+(b4+b11) \\ x(7)=(b1+b2)-(b4+b11) \end{cases}, \begin{cases} x(2)=(b0+b3)+(b9-b7) \\ x(5)=(b0+b3)-(b9-b7) \end{cases}$$
$$\begin{cases} x(1)=(b0-b3)+(b8-b6) \\ x(6)=(b0-b3)-(b8-b6) \end{cases}, \begin{cases} x(3)=(b1-b2)+(b10-b5) \\ x(4)=(b1-b2)-(b10-b5) \end{cases} \tag{16}$$

Therefore, the circuit structure is similar to Fig.4, which is an inverse transform between input and output.

4 Simulation and Performance Evaluation

Verilog hardware description language is used to complete the RTL design of 2-D DCT/IDCT unit based on CORDIC algorithm, and an image compression simulation verification platform is built. The test excitation adopts image Lena with size of 720×720. IDCT is performed on the data set processed by DCT to reconstruct the image. Then, the performance is evaluated based on FPGA,.

4.1 Simulation and Accuracy Evaluation of DCT Compression

The test excitation image Lena is divided into blocks of 8×8, and then input into the DCT unit for transformation. The Fig.6 (a) is the test excitation, and the Fig.6 (b) is the result of the DCT unit designed in this paper.

(a) (b)

Fig.6 (a) Test Excitation, (b) Result of DCT

In the process of DCT, different quantization coefficient tables are set, which can extract the low-frequency components with relatively concentrated energy in the DCT block. The quantization coefficient table is '1', and the components at the corresponding position can be saved. Otherwise, the components at the corresponding position are discarded. The proportion of '1' in the quantization coefficient table represents the compression ratio.

IDCT reconstruction is performed on images with compression ratios of 1, 15/64, and 1/64. The image reconstructed is shown in Fig.7.

(a) (b)

(c)

Fig.7 Result of IDCT (a) Compression Ratios of 1, (b) Compression Ratios of 15/64, (c) Compression Ratios of 1/64

The Peak Signal to Noise Ratio (PSNR) of the three images in Fig.7 is evaluated with the original image in Table 2. PSNR is an objective standard for evaluating images. It can be used to evaluate the compression effect of image compression algorithms and accurately and quantitatively represent the degree of image distortion.

Table 2 PSNR

Image	Compression Ratios	PSNR
(a)	1	51.99
(b)	15/64	43.65
(c)	1/64	29.97

The PSNR of the IDCT reconstructed image after the two-dimensional DCT uncompressed transform is 51.99 dB. Table 3 shows the comparison results of different architectures.

Table 3 PSNR of Various Architecture

Architecture	PSNR
Proposed	51.99
Liang[20]	44.48
Satpute[21]	33.95
Lee[22]	31.45

4.2 Evaluation of Area, Latency and Power Consumption

The area, latency and power consumption of the proposed DCT unit are evaluated. The design synthesizes on the Xilinx's Kintex-7 FPGA, and the reuslts are shown in Table 4.

Table 4 Area, Latency And Power Consumption

Architecture	Slice LUTs	Slice Registers	Latency (ns)	Power (mW)
Proposed	17602 (17.36%)	7085 (3.49%)	5.6	466

The resource utilization is 17.36% for Slice LUTs, 3.49% for Slice Registers. The maximum operating frequency can reach 172MHz, and it takes 28 clock cycles to complete a process of 2-D 8×8 samples, which is about 0.161μs. The architecture proposed by adaghiani et al.[23] and Subramanianp et al.[24] requires 0.386μs and 0.271μs, respectively. Compared with them, the proposed architecture is improved by 59.4% and 42.2% respectively.

6 Conclusion

This paper proposes a hardware implementation architecture of two-dimensional Discrete Cosine Transform (DCT) algorithm based on improved COordinate Rotation DIgital Computer (CORDIC), which reduces the computational complexity. The rotation matrix of CORDIC algorithm is used to replace the multiplication of cosine function in DCT, which is an approximate multiplier-free design and achieves high computational accuracy. The Peak Signal to Noise Ratio (PSNR) can reach 51.99 dB, and a comparison is made between the proposed architecture and other DCT architectures, which shows that the proposed architecture has the best performance. Finally, the circuit is synthesized based on FPGA. The pipeline latency of proposed architecture is 28 clock cycles, and the maximum operating frequency can reach 172 MHz. Experiments show that the proposed DCT structure based on Improved CORDIC is efficient and high accuracy, which can be applied in image and video compression.

References

[1] Phadatare P and Chavan P. Implementation of DCT and Fractal Compression Technique[C]. 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2021: 83-87.

[2] Messaoudi A. DCT-based compression algorithm using reduced adaptive block scanning for color image[C]. 2022 19th International Multi-Conference on Systems, Signals & Devices (SSD), Sétif, Algeria, 2022: 1951-1955.

[3] Chouhan A and Nigam M J. Double compression of JPEG image using DCT with estimated quality factor[C]. 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 2016: 1-3.

[4] Fung K T and Siu W S. Conversion between DCT coefficients and it coefficients in the compressed domain for H.263 to H.264 video transcoding[C]. IEEE International Conference on Image Processing 2005, Genova, Italy, 2005: III-57.

[5] Wu E P, Bui T A, Chen K, et al. Hardware Implementation of DCT/IDCT sharing for HEVC/MPEG Video Coding[C]. 2020 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2020: 1-3.

[6] Akman A and Cekli S. Design of Approximate Discrete Cosine Transform Architecture for Image Compression with HEVC Intra Prediction[C]. 2020 12th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 2020: 155-158.

[7] Akman A and Cekli S. Design of Approximate Discrete Cosine Transform Architecture for Image Compression with HEVC Intra Prediction[C]. 2020 12th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 2020: 155-158.

[8] Loeffler C, Ligtenberg A and Moschytz G S. Practical fast 1-D DCT algorithms with 11 ultiplications[C]. International Conference on Acoustics, Speech, and Signal Processing, Glasgow, UK, 1989(2): 988-991.

[9] Kitsos P, Voros N S, Dagiuklas T, et al. A high speed FPGA implementation of the 2D DCT for Ultra High Definition video coding[C]. 2013 18th International Conference on Digital Signal Processing (DSP), Fira, Greece, 2013: 1-5.

[10] Tran T D. The binDCT: fast multiplierless approximation of the DCT[J]. IEEE Signal Processing Letters, 2000, 7(6): 141-144.

[11] Cintra R J and Bayer F M. A DCT Approximation for Image Compression[J]. IEEE Signal Processing Letters, 2011, 18(10): 579-582.

[12] Yu S W and Swartzlander E E. A scaled DCT architecture with the CORDIC algorithm[J]. IEEE Transactions on Signal Processing, 2002, 50(1): 160-167.

[13] Sung T Y, Shieh Y S, Yu C W, et al. High-Efficiency and Low-Power Architectures for 2-D DCT and IDCT Based on CORDIC Rotation[C]. 2006 Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), Taipei, Taiwan, 2006: 191-196.

[14] Zhang J, Shi W, Zhou L, et al. A Low-Power and High-PSNR Unified DCT/IDCT Architecture Based on EARC and Enhanced Scale Factor Approximation[J]. IEEE Access, 2019(7): 165684-165691.

[15] Langelaar G C and Lagendijk R L. Optimal differential energy watermarking of DCT encoded images and video[J]. IEEE Transactions on Image Processing, 2001, 10(1): 148-158.

[16] Meher P K, Park S Y, Mohanty B K, et al. Efficient Integer DCT Architectures for HEVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(1): 168-178.

[17] Srivastava P K and Jakkani A K. FPGA Implementation of Pipelined 8×8 2-D DCT and IDCT Structure for H.264 Protocol[C]. 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 2018: 1-6.

[18] Teja G R, Sruthi R S, Tomar K S, et al. Verilog implementation of fully pipelined and multiplierless 2D DCT/IDCT JPEG architecture[C]. 2015 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 2015: 1-5.

[19] Volder J E. The CORDIC Trigonometric Computing Technique[J]. in IRE Transactions on Electronic Computers, 1959, 8(3): 330-334.

[20] Liang J and Tran T D. Fast multiplierless approximations of the DCT with the lifting scheme[J]. IEEE Transactions on Signal Processing, 2001, 49(12): 3032-3044.

[21] Satpute V R and Nawandar N K. CORDIC based novel energy-efficient approximate DCT architecture for error-resilient applications[C]. *2016* 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee, India, 2016: 655-660

[22] Lee M W, Yoon J H and Park J. Reconfigurable CORDIC-Based Low-Power DCT Architecture Based on Data Priority[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014, 22(5): 1060-1068.

[23] Sadaghiani A K and Ghanbari M. An Optimized Hardware Design for high speed 2D-DCT processor based on modified Loeffler architecture[C]. 2019 27th Iranian Conference on Electrical Engineering (ICEE), Yazd, Iran, 2019: 1476-1480.

[24] Subramanian P and Reddy A S C. VLSI Implementation of fully pipelined multiplierless 2D DCT/IDCT architecture for JPEG[C]. IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS, Beijing, China, 2010: 401-404.

Author Biographies

WU Hao received his B.Sc. degree in Microelectronics science and engineering from Soochow University in 2020. He is currently a M.Sc. candidate in University of Science and Technology of China (USTC). His main research interests include digital image processing and VLSI design.

E-mail: haow2020@mail.ustc.edu.cn

RU Zhanqiang received his M.Sc. degrees in Optical engineering from Changchun University of Science and Technology in 2010. From 2010 to 2019, he was a Research Assistant with Suzhou Institute of Nano-Tech and Nano-Bionics (SINANO), CAS. Since 2019, he has been a Senior Engineer with the SINANO, China. His research interests include laser beam scanning display technology and non-imaging optics design.

E-mail: zqru2008@sinano.ac.cn