

# Memory-Efficient and High-Performance 2-D DCT and IDCT Processors Based on CORDIC Rotation

TZE-YUN SUNG  
 Department of Microelectronics Engineering  
 Chung Hua University  
 707, Sec. 2, Wufu Road  
 Hsinchu, 30012, TAIWAN

*Abstract:* - Two-dimensional discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) have been widely used in many image processing systems. In this paper, efficient architectures with parallel and pipelined structures are proposed to implement  $8 \times 8$  DCT and IDCT processors. In which, only one bank of SRAM (64 words) and coefficient ROM (6 words) is utilized for saving the memory space. The kernel arithmetic unit, i.e. multiplier, which is demanding in the implementation of DCT and IDCT processors, has been replaced by simple adders and shifters based on the CORDIC algorithm. The proposed architectures for 2-D DCT and IDCT processors not only simplify hardware but also reduce the power consumption with high performances.

*Key-Words:* - DCT, IDCT, parallel and pipelined architecture, low-power, CORDIC.

## 1 Introduction

With the rapid growth of modern communication applications and computer technologies, image compression is increasingly in demand. Discrete cosine transform (DCT) has been widely used in the image compression task. Moreover, DCT is adopted by the JPEG, MPEG-4 and H.264 standards.

Conventionally, the double size fast Fourier transform (FFT) algorithm can be used to implement DCT. Nevertheless, FFT involves complex-valued computations. Specifically, for  $N$ -point DCT, the number of processor units required is  $2 \log 2N$  and the order of computation time is  $O(\log 2N + 1)$  by FFT. The VLSI chip implementations of DCT for real-time applications can be found in [1]-[8].

CORDIC (COordinate Rotation Digital Computer) is a well-known technique that was (and still is) widely used for the calculation of many elementary functions including sine and cosine functions. In this paper, the CORDIC approach to the implementation of fast DCT and IDCT is presented. The proposed CORDIC-based parallel and pipelined architectures for the development of two dimensional DCT and IDCT processors can simplify the hardware complexity and reduce the power consumption as well.

The remainder of this paper proceeds as follows. In Section 2, the CORDIC algorithm is reviewed briefly. In Section 3, fast and efficient CORDIC-

based 2-D DCT and IDCT algorithms are presented. The implementations of the proposed low-power, parallel and pipelined architectures for 2-D DCT and IDCT processors are given in Section 4. Finally, conclusion can be found in Section 5.

## 2 Review of CORDIC Algorithm

The basic CORDIC algorithm is given by [9]-[10]

$$x_{i+1} = x_i - \sigma_i 2^{-i} y_i \tag{1}$$

$$y_{i+1} = y_i + \sigma_i 2^{-i} x_i \tag{2}$$

$$z_{i+1} = z_i - \sigma_i \alpha_i \tag{3}$$

where  $i=0, 1, 2, \dots, n-1$ , and

$$\alpha_i = \arctan(2^{-i}) \tag{4}$$

In the  $i^{th}$  micro-rotation, the direction of rotation denoted by  $\sigma_i$  is determined by  $sign(z_i)$  with  $z_n \rightarrow 0$  in the rotation mode;  $\sigma_i = -sign(x_i) \cdot sign(y_i)$  with  $y_n \rightarrow 0$  in the vectoring mode; and the corresponding scale factor  $k_i = \frac{1}{\sqrt{1 + \sigma_i^2 2^{-2i}}}$ . After  $n$  micro-rotations, the product of all the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k_i = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + \sigma_i^2 2^{-2i}}} = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + 2^{-2i}}} \tag{5}$$

One may take the iteration sequence:  $\{0, 0, 0, 1, 2, \dots, n\}$  for the CORDIC algorithm in the circular coordinate system to expand the convergence range of angles as follows.

$$\theta_{\max} = \arctan(2^{-n}) + 2 \cdot \arctan(2^0) + \sum_{j=0}^n \arctan 2^{-j} \quad (6)$$

$$\cong 3.3141(189^\circ) > 180^\circ$$

Thus, the convergence range of angles is expanded to  $\pm 180^\circ$ , and the input angle can be unlimited [11]-[12].

### 3 The CORDIC-Based DCT and IDCT Algorithm

The  $N$ -point 1-D DCT is defined as

$$Y(m) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sqrt{2} K_m \cos\left[\frac{(2n+1)m\pi}{2N}\right] \cdot x(n) \quad (7)$$

where  $m = 0, \dots, N-1$ ,  $K_m = \frac{1}{\sqrt{2}}$  for  $m = 0$ , and

$K_m = 1$  for  $m > 0$ .

For image applications, a separable 2-D DCT can be obtained by using the tensor product of two 1-D DCTs. Specifically, the  $M \times N$ -point 2-D DCT is defined as

$$Z(u, v) = \frac{2 \cdot c(u)c(v)}{\sqrt{M \cdot N}} \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m, n) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \quad (8)$$

where  $u = 0, \dots, M-1, v = 0, \dots, N-1$ ,  $c(k) = \frac{1}{\sqrt{2}}$

for  $k = 0$ , and  $c(k) = 1$  for  $k > 0$ . Equation (8) can be rewritten by

$$Z(u, v) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} \sqrt{2} c(u) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \left\{ \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \sqrt{2} c(v) \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \cdot x(m, n) \right\} \quad (9)$$

For  $8 \times 8$  DCT, let

$$\mathbf{T} = \frac{1}{\sqrt{8}} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a & c & d & f & -f & -d & -c & -a \\ b & e & -e & -b & -b & -e & e & b \\ c & -f & -a & -d & d & a & f & -c \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ d & -a & f & c & -c & -f & a & -d \\ e & -b & b & -e & -e & b & -b & e \\ f & -d & c & -a & a & -c & d & -f \end{bmatrix} \quad (10)$$

where  $a = \sqrt{2} \cos\left(\frac{\pi}{16}\right)$ ,  $b = \sqrt{2} \cos\left(\frac{\pi}{8}\right)$ ,

$c = \sqrt{2} \cos\left(\frac{3\pi}{16}\right)$ ,  $d = \sqrt{2} \cos\left(\frac{5\pi}{16}\right)$ ,

$e = \sqrt{2} \cos\left(\frac{3\pi}{8}\right)$ , and  $f = \sqrt{2} \cos\left(\frac{7\pi}{16}\right)$ .

The transform coefficients  $Z(u, v)$  of  $8 \times 8$  DCT can be grouped into an array denoted by  $\mathbf{Z}$ , which can be written by

$$\mathbf{Z} = \mathbf{T}\mathbf{Y}^t \quad (11)$$

where  $\mathbf{Y} = \mathbf{T}\mathbf{X}^t$ . Thus, the computation of separable 2-D DCT can be obtained by using 1-D DCT computation as follows.

$$2\text{-D DCT}(\mathbf{X}) = 1\text{-D DCT}((1\text{-D DCT}(\mathbf{X}))^t) \quad (12)$$

Similarly, a separable  $M \times N$ -point 2-D IDCT can be obtained, which is given by

$$x(m, n) = \frac{2 \cdot c(u)c(v)}{\sqrt{M \cdot N}} \cdot \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} Z(u, v) \cdot \cos\left[\frac{(2m+1)u\pi}{2M}\right] \cdot \cos\left[\frac{(2n+1)v\pi}{2N}\right] \quad (13)$$

where  $u = 0, \dots, M-1, v = 0, \dots, N-1$ ,  $c(k) = \frac{1}{\sqrt{2}}$  for

$k = 0$ , and  $c(k) = 1$  for  $k > 0$ .

The 2-D IDCT computation using 1-D IDCT computation is as follows.

$$2\text{-D IDCT}(\mathbf{Z}) = 1\text{-D IDCT}((1\text{-D IDCT}(\mathbf{Z}))^t) \quad (14)$$

In which,  $\mathbf{X} = \mathbf{T}^t \mathbf{Z} \mathbf{T}$ ,  $\mathbf{Y} = \mathbf{T}^t \mathbf{Z}^t$ , and therefore

$$\mathbf{X} = \mathbf{T}^t \mathbf{Y}^t \quad (15)$$

#### 3.1 Fast 1-D DCT Algorithm

Matrix  $\mathbf{T}$  defined by equation (10) can be further decomposed to obtain a fast algorithm for 1-D DCT. Specifically, the fast 8-point DCT is given by

$$\begin{bmatrix} Y(0) \\ Y(2) \\ Y(4) \\ Y(6) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ b & e & -e & -b \\ 1 & -1 & -1 & 1 \\ e & -b & b & -e \end{bmatrix} \begin{bmatrix} x(0) + x(7) \\ x(1) + x(6) \\ x(2) + x(5) \\ x(3) + x(4) \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} Y(1) \\ Y(3) \\ Y(5) \\ Y(7) \end{bmatrix} = \begin{bmatrix} a & -c & d & -f \\ c & f & -a & d \\ d & a & f & -c \\ f & d & c & a \end{bmatrix} \begin{bmatrix} x(0) - x(7) \\ -x(1) + x(6) \\ x(2) - x(5) \\ -x(3) + x(4) \end{bmatrix} \quad (17)$$

Figure 1 shows the data flow of 8-point DCT, where the blocks named CORDIC(2) and CORDIC(5) are constructed by the same structure with rotation angle  $\pi/16$ ; the blocks named CORDIC(3) and CORDIC(4) are of the same structure with rotation angle  $5\pi/16$ .

### 3.2 Fast 1-D IDCT Algorithm

Similarly, the fast 8-point IDCT can be obtained by further decomposing Matrix  $T$ , which is given by

$$\begin{bmatrix} x(0) \\ x(7) \\ x(4) \\ x(3) \end{bmatrix} = \begin{bmatrix} 1 & b & e & f & a & c & d \\ 1 & b & e & -f & -a & -c & -d \\ 1 & -b & -e & a & f & c & -d \\ 1 & -b & -e & -a & -f & -a & c \end{bmatrix} \begin{bmatrix} Y(0)+Y(4) \\ Y(2) \\ Y(6) \\ Y(1) \\ Y(7) \\ Y(3) \\ Y(5) \end{bmatrix} \quad (18)$$

$$\begin{bmatrix} x(1) \\ x(5) \\ x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} 1 & e & -b & c & -d & -f & -a \\ 1 & -e & -b & -d & -c & a & -f \\ 1 & -e & b & d & c & -a & f \\ 1 & e & -b & -c & d & f & a \end{bmatrix} \begin{bmatrix} Y(0)-Y(4) \\ Y(2) \\ Y(6) \\ Y(7) \\ Y(1) \\ Y(3) \\ Y(5) \end{bmatrix} \quad (19)$$

Figure 2 shows the data flow of 8-point IDCT, where the blocks named R0 and R2 are constructed by the same structure with rotation angles  $\pi/16$  and  $5\pi/16$ , and block R1 involves a rotation of angle  $6\pi/16$ .

## 4 The Proposed 2-D DCT and IDCT Processors

Multiplication operation, which is demanding in the computation of both DCT and IDCT, can be avoided by using the CORDIC-based processor.

Based on equations (11) and (15), an efficient parallel-pipelined architecture has been developed for both 2D DCT and IDCT. Figure 3 shows the proposed architecture for  $8 \times 8$  DCT and IDCT processors, where one SRAM bank (64 words), two 8-point DCT/IDCT processors, and the control unit are involved. The 8-point 1-D DCT/IDCT input-processor, which is denoted by P1, writes the intermediate result into the row and column of SRAM bank alternately. The 8-point 1-D DCT/IDCT output-processor, which is denoted by P2, reads data from the column and row of SRAM bank alternately and outputs the final result. Figure 4 shows the finite state machine (FSM) of the control unit.

The implemented 8-point DCT/IDCT processor involves five CORDIC processors that are obtained by using the CORDIC arithmetic. Figure 5 shows the proposed 8-point DCT processor, and Figure 6 shows the proposed 8-point IDCT processor. It is

noted that the transformation matrices of DCT and IDCT are column symmetry and row symmetry, respectively, the shuffle structures are simplified, and no multipliers are required.

In Figure 3, the latency of the constituent 1-D DCT/IDCT processors is 8 clocks, the hardware complexity is  $O(N \cdot \log_2 N)$ , and the throughput is 8 outputs per cycle. Since no multiplier is utilized, many desirable properties such as small area, low-power and high throughput can be achieved. Table 1 shows the comparison to the commonly used architectures of [1]-[8].

The proposed parallel-pipelined architecture for 2-D DCT and IDCT processors have been written in Verilog<sup>®</sup> and synthesized by TSMC 0.18  $\mu\text{m}$  1P6M CMOS cell libraries [13]. The core sizes and power consumptions can be obtained from the reports of Synopsys<sup>®</sup> design analyzer and PrimPower<sup>®</sup> [14], respectively. The reported core sizes of the implemented 2-D DCT and IDCT processors are  $2372 \times 2372 \mu\text{m}^2$  and  $2396 \times 2396 \mu\text{m}^2$ , and the power dissipations are 127.7 mW at 1.8V with clock rate of 34.4MHz and 116.7 mW at 1.8V with clock rate of 35.7MHz, respectively. Figures 7 and 8 show the layout views of the implemented 2-D DCT and IDCT processors, respectively. The original  $512 \times 512$  Lena image is shown in Figure 9; the reconstructed image is shown in Figure 10. Through the proposed architectures for 32-bit fixed-point DCT/IDCT, the peak-signal-to-noise ratio (PSNR) of the reconstructed image is 44.6dB. The proposed 2-D DCT/IDCT processors have been applied to various images with great satisfactions.

## 5 Conclusion

By taking into account the symmetry properties of the fast DCT and IDCT algorithms, high efficiency architectures with parallel and pipelined structures have been proposed to implement DCT and IDCT processors. For image applications, a separable 2-D DCT/IDCT can be obtained by using the tensor product of two 1-D DCT/IDCT operations. The proposed 2-D DCT/IDCT processor is composed of two successive 1-D DCT/IDCT kernels with single memory bank. In the constituent 1-D DCT/IDCT processors, the CORDIC algorithm with rotation mode in the circular coordinate system has been utilized for the arithmetic unit (AU) involved, i.e. the multiplication computation. The proposed DCT/IDCT architectures are not only regularly structured but also highly scalable and flexible as

well. The DCT and IDCT processors are reusable IPs that have been implemented in various processes, and in combination with an efficient use of the hardware resources available in the target systems leads to various performances, area and power consumption trade-offs. The proposed 2-D DCT and IDCT processors are much suited to the applications of JPEG, MPEG-4 and H.264 standards.

References:

[1] Y. P. Lee, T. H. Chen, L. G. Chen, C. W. Ku, A Cost-Effective Architecture for 8×8 two-dimensional DCT/IDCT Using Direct Method, *IEEE Trans. on Circuits Systems for Video Technology*, Vol. 7, No. 1, June 1997, pp.459-467.

[2] Y. T. Chang, C. L. Wang, New Systolic Array Implementation of the 2-D Discrete Cosine Transform and Its Inverse, *IEEE Trans. on Circuits Systems for Video Technology*, Vol. 5, No. 1, April 1995, pp. 150-157.

[3] S. F. Hsiao, W. R. Shiue, "A New Hardware-Efficient Algorithm and Architecture for Computation of 2-D DCTs on a Linear Array," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 11, Nov. 2001, pp.1149-1159.

[4] S. F. Hsiao, J. M. Tseng, New Matrix Formulation for Two-Dimensional DCT/IDCT Computation and its Distributed-Memory VLSI Implementation, *IEE Proc.-Vis. Image Signal Process*, Vol. 149, No. 2, April 2002, pp.97-107.

[5] S. F. Hsiao, Y. H. Hu, T. B. Juang, C. H. Lee, Efficient VLSI Implementations of Fast Multiplierless Approximated DCT Using Parameterized Hardware Modules for Silicon Intellectual Property Design, *IEEE Trans. on Circuits and Systems, Part-I*: Vol. 52, No. 8, 2005, pp. 1568-1579.

[6] T.-Y. Sung, Y.-H. Sung, A Novel Implementation of Cost-Effective Parallel-Pipelined 8×8 DCT Processor, *The Fourth IEEE Asia-Pacific Conference on Advanced System Integrated Circuits*, Fukuoka, Japan, August 3-5, 2004, pp.200-203.

[7] T.-Y. Sung, Y.-S. Shieh, M.-J. Sun, A High-Throughput and Memory-Efficiency 2-D DCT Architecture Based on CORDIC Rotation, *The 23rd Workshop on Combinatorial Mathematics and Computation Theory*, Taiwan, April 28~29, 2006, pp.369-372.

[8] T.-Y. Sung, M.-J. Sun, Y.-S. Shieh, H.-C. Hsin, Memory-Efficiency and High-Speed Architectures for Forward and Inverse DCT with Multiplierless Operation, *2006 IEEE Pacific-Rim Symposium on Image and Video Technology*, Hsinchu, Taiwan, December 11-13, 2006, pp.802-811.

[9] J. E. Volder, The CORDIC Trigonometric Computing Technique, *IRE Trans. on Electronic Computers*, Vol. EC-8, 1959, pp.330-334.

[10] J. S. Walther, A Unified Algorithm for Elementary Functions, *Spring Joint Computer Conference Proceedings*, Vol. 38, 1971, pp. 379-385.

[11] X. Hu, R. G. Harber, S. C. Bass, Expanding the Range of the Convergence of the CORDIC Algorithm, *IEEE Trans. on Computers*, Vol. 40, No. 1, 1991, pp.13-21.

[12] T.-Y. Sung, Y.-H. Sung, The Quantization Effects of CORDIC Arithmetic for Digital Signal Processing Applications, *The 21<sup>st</sup> Workshop on Combinatorial Mathematics and Computation Theory*, Taiwan, May 21~22, 2004, pp.16-25.

[13] *TSMC 0.18 μm CMOS Design Libraries and Technical Data, v.3.2*, Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan, and National Chip Implementation Center (CIC), National Science Council, Hsinchu, Taiwan, R.O.C., 2006.

[14] Synopsys products, <http://www.synopsys.com/products>.

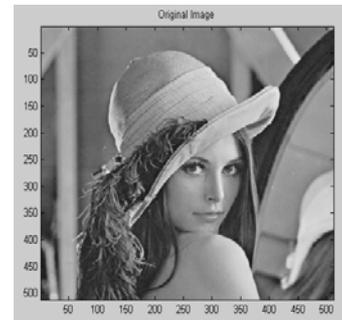


Fig.9 Original image

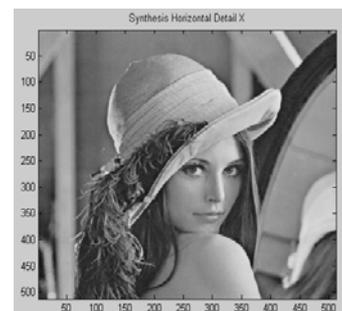


Fig.10 Reconstructed image

Table 1 Comparison of the proposed architecture to other commonly used architectures

8x8 2-D DCT/IDCT	Lee [1] DCT/IDCT	Chang [2] DCT/IDCT	Hsiao [3] DCT	Hsiao [4] DCT/IDCT	Hsiao [5] DCT/IDCT	Sung [6]-[8] DCT/IDCT	This Work DCT/IDCT
Real-multipliers	28	64	-	-	-	-	-
CORDIC processors	-	-	-	-	3	5	5
Real-adders	134	88	-	10	14	18	18
Complex-multipliers	-	-	3	3	-	-	-
Complex-adders	-	-	9	-	-	-	-
Delay elements (Words)	256	114	-	171	-	-	-
Memory (Words)	~384	~200	~370	-	-	134	70
Hardware complexity (AUs)	$O(N\log N)$	$O(N^2)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(N - \log N)$	$O(N - \log N)$
Throughput (outputs/cycle)	16	8	2	2	2	8	8
Hardware utilization (100%)	no	no	no	no	no	yes	yes
Pipelining	no	no	no	no	yes	yes	yes
Parallelism	yes	yes	yes	yes	yes	yes	yes

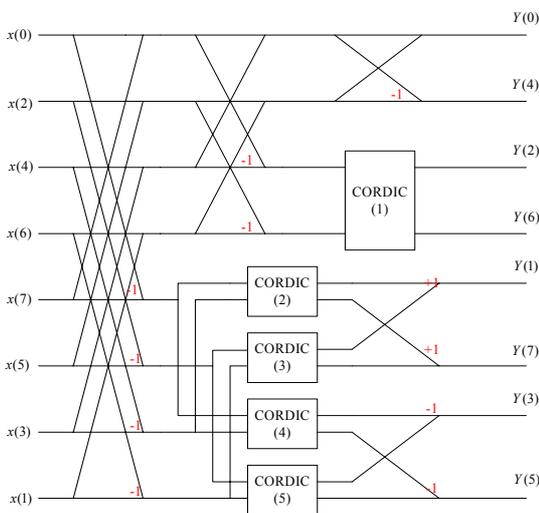


Fig.1. Data flow of 8-point DCT

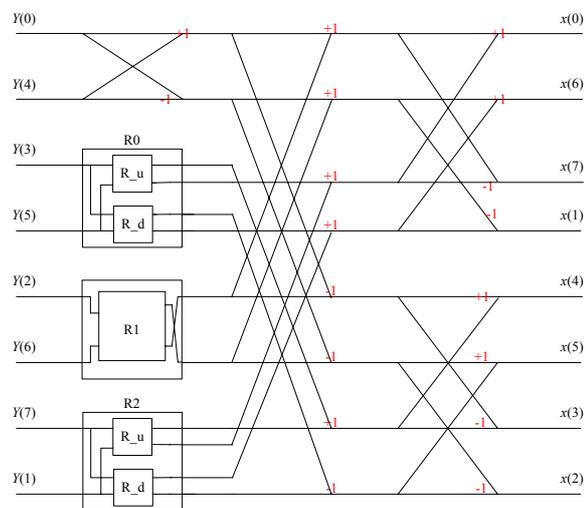


Fig.2. Data flow of 8-point IDCT

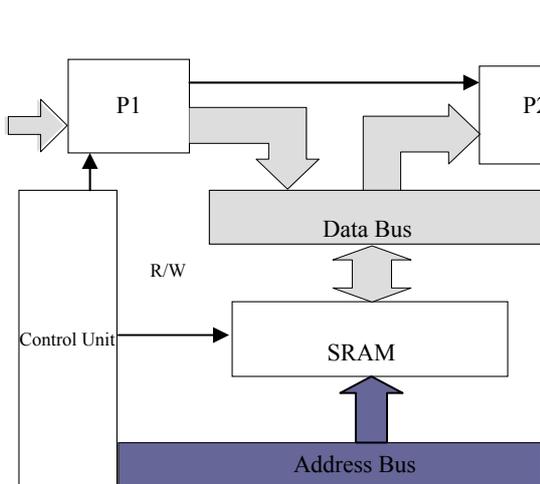


Fig.3. The proposed architecture for 2-D DCT/IDCT processor (P1 and P2: 1-D DCT/IDCT processor)

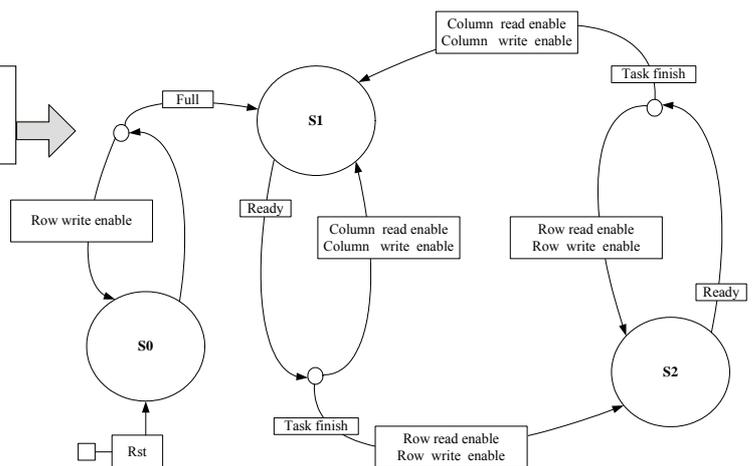


Fig.4. The finite state machine (FSM) of the control unit

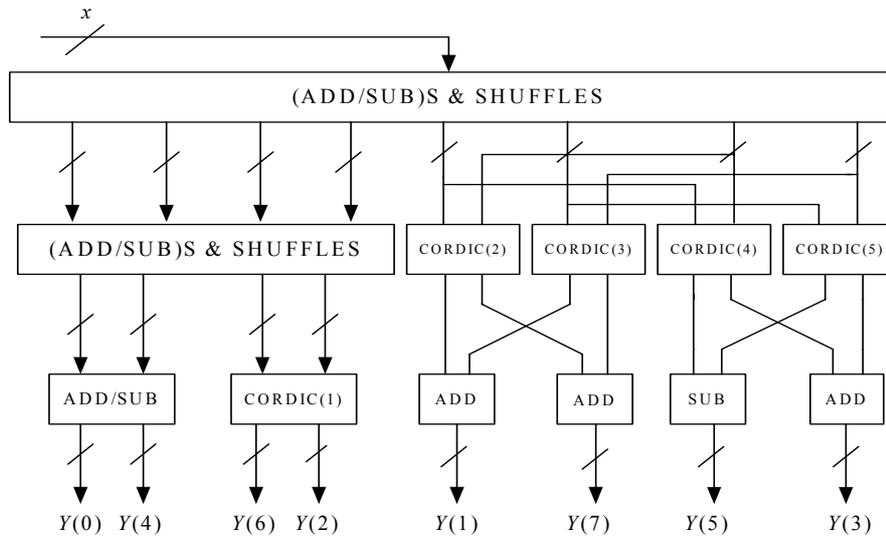


Fig.5. The proposed 1-D 8-point DCT processor

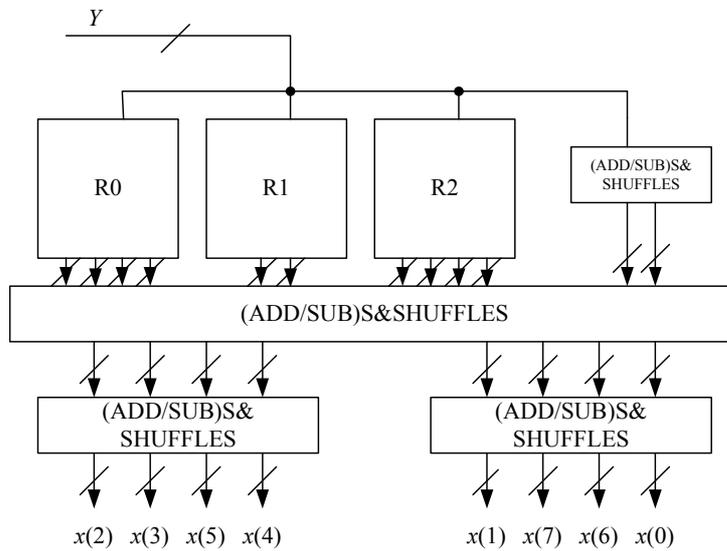


Fig.6. The proposed 1-D 8-point IDCT processor

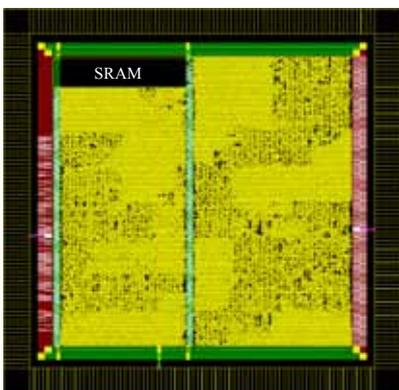


Fig.7. The layout view of the implemented 2-D DCT processor

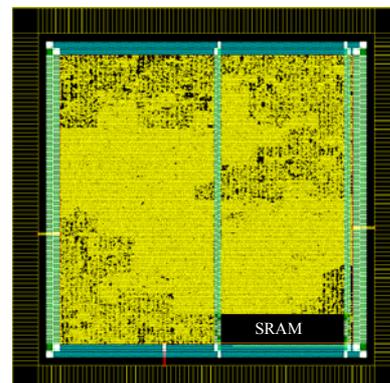


Fig.8. The layout view of the implemented 2-D IDCT processor