VLSI Implementation of Memory-Efficiency Multiplierless DCT and IDCT Processors

Tze-Yun Sung[1], Yaw-Shih Shieh[1] and Hsi-Chin Hsin[2]
[1]*Dept. of Microelectronics Engineering, Chung Hua University, Hsinchu, Taiwan 30012, R.O.C.*
[2]*Dept. of Computer Science and Information Engineering, National Formosa University, Hu-Wei, Yunlin, Taiwan 63208, R.O.C.*
E-mail: bobsung@chu.edu.tw

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 8×8 *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 The kernel arithmetic unit, i.e. multiplier, which is demanding in the implementation of DCT and IDCT processors, has been replaced by simple adders and shifters based on the CORDIC algorithm.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.*

1: INTRODUCTION

With the rapid growth of modern communication applications and computer technologies, image compression is increasingly in demand. From the compression point of view, transform coding is superior to linear prediction coding. Walsh-Hadamard transform is the simplest one, in which the computations involved in the kernel matrix are only additions and subtractions [1]. As cosine transform approximates to the optimal Karhunen-Loeve transform that is much more complicated in practice [2], the discrete cosine transform (DCT) has been widely used in the image compression task. Moreover, DCT has been adopted by the JPEG standard.

Conventionally, the double size fast Fourier transform (FFT) algorithm can be used to implement DCT. However, FFT involves complex-valued computations. Specifically, for *N*-point DCT, the required number of processor units is $2\log 2N$ and the order of computation time is $O(\log 2N+1)$ while applying FFT. Several fast computation algorithms were thus proposed with discussions [3]-[12], and the VLSI chip implementations of DCT for real-time applications can be found in [13]-[23].

In this paper, the CORDIC-based approach to the implementation of fast DCT and IDCT is proposed. In Section 2, the CORDIC algorithm is described briefly. In Section 3, both the CORDIC-based fast 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 are given in Section 4. Finally, conclusion can be bound in Section 5.

2: REVIEW OF CORDIC ALGORITHM

COordinate Rotation DIgital Computer (CORDIC) is a well-known algorithm that evaluates many fundamental functions in the iterative manner [24]-[25]. As the hardware implementation of CORDIC may require only simple adders and shifters, it has received a lot of attention. A rotation of angle θ in the circular coordinate system can be obtained by performing a sequence of micro-rotations successively. Specifically, a vector can be rotated by the use of a sequence of pre-determined step-angles. The basic CORDIC algorithm in the circular coordinate system is as follows.

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

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

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

where i=0, 1,2, …., n-1, and

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

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

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

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

$$\theta_{\max} = \arctan(2^{-n}) + 2\cdot\arctan(2^{0}) + \sum_{j=0}^{n}\arctan 2^{-j} \cong 3.3141(189^\circ) > 180^\circ \quad (6)$$

Thus, the convergence range of angles is expanded to $\pm 180^\circ$, and the input angle can be unlimited [26]-[27].

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 $m = 0,...., N-1$, $K_m = \frac{1}{\sqrt{2}}$ 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 $u = 0,....,M-1, v = 0,....,N-1$, $c(k) = \frac{1}{\sqrt{2}}$ 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 8×8 DCT, let

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

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 8×8 DCT can be arranged and grouped into an array denoted by $\boldsymbol{Z}$, which can be written by

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

where $\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{X}^t$. As a result, the separable 2-D DCT computation can be obtained by using 1-D DCT computations as follows.

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

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 $\mu = 0,....,M-1, \nu = 0,....,N-1$, $c(k) = \frac{1}{\sqrt{2}}$ for k=0, and $c(k) = 1$ for k>0.

Thus, the 2-D IDCT computation using 1-D IDCT computations is as follows.

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

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

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

3.1: FAST 1-D DCT ALGORITHM

Equation (10) can be further decomposed to obtain a fast algorithm for 1-D DCT [28]. Specifically, 8-point fast DCT is as follows.

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

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

Based on equations (16) and (17), the data flow of 8-point DCT can be determined, which is shown in Figure 1. It is noted that CORDIC(2) and CORDIC(5) involved in Figure 1 have the same structure with rotation angle of $\frac{\pi}{16}$; CORDIC(3) and CORDIC(4) have the same structure with rotation angle of $\frac{5\pi}{16}$.

3.2: FAST 1-D IDCT ALGORITHM

By further decomposing equation (10), 8-point fast IDCT can be obtained, which is given by

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

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

Based on equations (18) and (19), the data flow of 8-point IDCT can be determined, which is shown in Figure 2. It is noted that the processor-R0 (with rotation angles of $\frac{5\pi}{16}$ and $\frac{\pi}{16}$) and the proceesor-R2 (with rotation angles of $\frac{\pi}{16}$ and $\frac{5\pi}{16}$) have the same structure; and the processor-R1 rotates by angle of $\frac{6\pi}{16}$.

3.3: THE PROPOSED CORDIC-BASED 2-D DCT AND IDCT ARCHITECTURES

Multiplication is the key operation for both DCT and IDCT. In the CORDIC-based processor with rotation mode in the circular coordinate system, multipliers of DCT and IDCT can be replaced by simple shifters and adders. Moreover, the arithmetic unit (AU) obtained by the use of double-rotation CORDIC algorithm has been taken into account to develop fast DCT and IDCT architectures. In comparison to the conventional CORDIC-based arithmetic unit, the proposed double-rotation CORDIC-based arithmetic unit can improve the latency more than 30% [29]. Thus, the hardware of arithmetic unit can be significantly saved and low-power consumption can be achieved. The overall relative error is less than 10^{-3} provided that the length of registers is 16 bits, and the number of micro-iterations in the double-rotation CORDIC processor is set to 4 [27].

4: THE PROPOSED 2-D DCT AND IDCT PROCESSORS

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 8×8 DCT and IDCT processors. In which, one SRAM bank (64 words), two 8-point DCT/IDCT processors, two multiplexers and control unit are involved. Specifically, 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 raw of SRAM bank alternately and outputs the final result. The control unit manages the data flow and arranges the timing for 2-D operations. Figure 4 shows the finite state machine (FSM) of control unit.

4.1: IMPLEMENTATION OF THE PROPOSED 1-D DCT AND IDCT PROCESSORS

The implemented 8-point DCT/IDCT processor utilizes five CORDIC processors obtained by using the double-rotation CORDIC arithmetic [29]. Figure 5 and 6 show the proposed 8-point DCT processor and IDCT processor, respectively. As the transformation matrices involved in 1-D DCT and IDCT are column symmetry and row symmetry, respectively, the shuffle structures of DCT/IDCT processor are therefore simplified, and no multipliers are needed.

4.2: IMPLEMENTATION OF THE PROPOSED 2-D DCT AND IDCT PROCESSORS

Figure 3 shows the proposed 2-D DCT/IDCT processor. In which, the latencies of the constituent 1-D DCT/IDCT processors are 64 clocks, hardware complexity is O(N-$\log_2 N$), and the throughputs are 8 outputs per cycle. As no multiplier is utilized in the proposed architecture, many desirable properties such as small area, low-power and high throughput are achieved. Table 1 shows the comparison of the proposed 2-D DCT/IDCT architecture to other commonly used architectures [9]-[13].

The proposed parallel-pipelined architectures for 2-D DCT and IDCT processors have been written in Verilog® and synthesized by TSMC 0.18 μm 1P6M CMOS cell libraries [30]. Their core sizes and power consumptions can be obtained from the reports of Synopsys® design analyzer and PrimPower® [31], respectively. The reported core sizes of 2-D DCT and IDCT processor are $2372 \times 2372\ \mu m^2$ and $2396 \times 2396\ \mu m^2$,respectively, and the power dissipations for 2-D DCT and IDCT processors are 127.7 mW at 1.8V with clock rate of 34.4MHz and 116.7 mW at 1.8V with clock rate 35.7MHz, respectively. Due to the limitation of paper size, the reports and layout view of the proposed 2-D IDWT processor are not presented here. Figure 7 and 8 show the layout views of the implemented 2-D DCT processor and IDCT processor, respectively. The proposed 2-D

DCT and IDCT processors are much suited to the applications of both JPEG and MPEG standards.

5: CONCLUSION

By taking into account the symmetry properties of the fast DCT/IDCT algorithm, high efficiency architectures with a parallel-pipelined structure 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. Thus, the proposed 2-D DCT/IDCT processor is composed of two successive 1-D DCT/IDCT kernels. In the constituent 1-D DCT/IDCT processors, the double-rotation CORDIC algorithm with rotation mode in the circular coordinate system has been utilized for the arithmetic unit (AU) of both DCT and IDCT, i.e. the multiplication computation. The proposed DCT/IDCT architectures are not only regularly structured but also highly scalable and flexible as well. Thus, they are much suited to VLSI implementation with design trade-offs.

REFERENCES

1. Elliott, D. F., Kao K. R., "Fast Transforms Algorithms, Analysis, Applications," Chapter 8, Walsh-Hadamard Transform, Prentice-Hall, (1982), 301-303.
2. Clarke, R. J., "Relation between the Karhenen Loeve and Cosine Transform," IEEE Proceedings, Part F, Vol. 128, No. 6, (1981), 359-360.
3. Narasimha, M. J., Peterson, A. M., "On the Computation of the Discrete Cosine Transform," IEEE Transactions on Communications, Vol. 26, No. 6, June 1978, pp. 934-936.
4. Haralick, R. M. "A Storage Way to Implement the Discrete Cosine Transform," IEEE Transactions on Computers, (1976), 764-765.
5. Chen, W. H., Smith, C. H., Fralick, S. C., "Fast Computational Algorithm for the Discrete Cosine Transform," IEEE Transactions on Communications, Vol. 25, No. 9, (1977), 1004-1009.
6. Sung, T. Y., "VLSI Parallel and Distributed Computation Algorithms for DCT Processors," Proceedings IEEE International Phoenix Conference on Computer and Communications, Scottsdale, Arizona, USA, (1990), 121-125.
7. Sung, T. Y., "VLSI Parallel and Distributed Processing Algorithms for Multidimensional Discrete Cosine Transforms," 1990 A Two-Track International Conference on Databases, Parallel Architectures, and their Applications, Miami Beach, Florida, USA, (1990), 36-39.
8. Sung, T. Y., "Novel Parallel VLSI Architectures for Discrete Cosine Transforms," Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, USA, (1990), 998-1001.
9. Lee, Y. P., Chen, T. H., Chen, L. G., Ku, C. W.," A Cost-Effective Architecture for 8×8 two-dimensional DCT/IDCT Using Direct Method," IEEE Transactions on Circuits Systems for Video Technology, Vol. 7, No. 1, (1997), 459-467.
10. Chang, Y. T., Wang, C. L., "New Systolic Array Implementation of the 2-D Discrete Cosine Transform and Its Inverse," IEEE Transactions on Circuits Systems for Video Technology, Vol. 5, No. 1, (1995), 150-157.
11. Hsiao, S. F., Shiue, W. R., "A New Hardware-Efficient Algorithm and Architecture for Computation of 2-D DCTs on a Linear Array," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, (2001), 1149-1159.
12. Hsiao, S. F., Tseng, J. M., "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, (2002), 97-107.
13. Hsiao, S. F. , Hu, Y. H., Juang, T. B., Lee, C. H., "Efficient VLSI Implementations of Fast Multiplierless Approximated DCT Using Parameterized Hardware Modules for Silicon Intellectual Property Design," IEEE Trans. Circuits and Systems, Part-I: Regular Papers, Vol. 52, No. 8, (2005), 1568-1579.
14. Srinvasan, V., Liu, K. J. R., "VLSI Design of High-Speed Time-Recursive 2-D DCT/IDCT Processor for Video Applications," IEEE Transactions on Circuits Systems for Video Technology, Vol. 6, No. 1, (1996), 87-96.
15. Kuroda, T., "A 0.9-V, 150-MHz, 10-mW, 4mm2, 2-D Discrete Cosine Transform Core Processor with Variable Threshold-Voltage(VT) Scheme," IEEE Journal of Solid-States Circuits, Vol. 31, No. 11, (1996), 1770-1778.
16. Rambaldi, R., Uguzzoni, A., Guerrieri, R., "A 35 μ W 1.1 V Gate Array 8×8 IDCT Processor for Video-Telephony," Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, (1998), 2993-2996.
17. Chen, T. H., "A Cost-Effective 8×8 2-D IDCT Core Processor with Folded Architecture," IEEE Transactions on Consumer Electronics, Vol. 45, No.2, (1999), 333-339.
18. Sung, T. Y., Sung, Y. H., "A Novel Implementation of Cost-Effective Parallel- Pipelined 8×8 DCT Processor," The Fourth IEEE Asia-Pacific Conference on Advanced System Integrated Circuits (AP-ASIC) 2004, Fukuoka, Japan, (2004), 200-203.
19. Hu, Y. H., Wu, Z., "An Efficient CORDIC Array Structure for the Implementation of Discrete Cosine Transform", IEEE Transactions on Signal Processing, Vol. 43, No. 1, (1995), 331-.336.
20. Jeong, H., Kim, J, Cho, W. K., "Low-Power Multiplierless DCT Architecture Using Image Data Correlation," IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, (2004), 262-267.
21. Gong, D., He, Y, Gao, Z., "New Cost-Effective VLSI Implementation of a 2-D Discrete Cosine Transform and Its Inverse", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, No. 4, (2004), 405-415.
22. Dimitrov, V., Wahid, K., Jullien, G., "Multiplication-Free 8×8 2D DCT Architecture Using Algebraic Integer Encoding", Electronics Letters, Vol. 40, No. 20, (2004).
23. Alam, M., Badawy, W., Jullien, G., "A New Time Distributed DCT Architecture for MPEG-4 Hardware Reference Model", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 5, (2005), 726-730.
24. Volder, J. E., "The CORDIC Trigonometric Computing Technique," IRE Transactions on Electronic Computers, Vol. EC-8, (1959), 330-334.
25. Walther, J. S., "A Unified Algorithm for Elementary Functions," Spring Joint Computer Conference Proceedings, Vol.38, (1971), 379-385.

26. Hu, X., Harber, R. G., Bass, S. C., "Expanding the range of the Convergence of the CORDIC Algorithm", IEEE Transactions on Computers, Vol. 40, No. 1, (1991), 13-21.
27. Sung, T. Y., Sung, Y. H., "The Quantization Effects of CORDIC Arithmetic for Digital Signal Processing Applications", The 21st Workshop on Combinatorial Mathematics and Computation Theory, Taiwan, (2004), 16-25.
28. Sung, T. Y., "A Memory-Efficient and High-Speed Split-Radix FFT/IFFT Processor Based on Pipelined CORDIC Rotations," to appear in IEE Proceedings – Vision, Image and Signal Processing, (2006).
29. Sung, T. Y., Chen, C. S., Shih, M. C., "The Double Rotation CORDIC Algorithm: New Results for VLSI Implementation of Fast Sine/Cosine Generation," 2004 International Computer Symposium (ICS-2004), Taipei, Taiwan, (2004), 1285-1290.
30. "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 Applied Research Labs., Hsinchu, Taiwan, R.O.C., (2006).
31. Synopsys FPGA Express, http://www. synopsys.com/ products.

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

8x8 2-D DCT/IDCT	Lee[9]	Chang[10]	Hsiao[11]	Hsiao[12]	Hsiao[13]	This Work
Real-adders	134	88	-	10	14	36
Real-multipliers	28	64	-	-	-	-
Rotators	-	-	-	-	3	5
Complex-multipliers	-	-	3	3	-	-
Delay elements (Words)	256	114	-	171	-	-
Memory(Words)	~384	~200	~370	-	-	70
Hardware complexity (AUs)	*O(NlogN)*	*O(N^2)*	*O(logN)*	*O(logN)*	*O(logN))*	*O(N-logN)*
Throughput (outputs/cycle)	16	8	2	2	2	8
Pipelinability	no	no	no	no	yes	yes
Parallelism	yes	yes	yes	yes	yes	yes

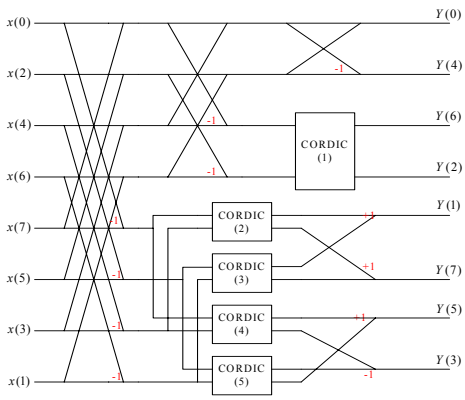

Fig. 1. Data flow of 8-point 1-D DCT

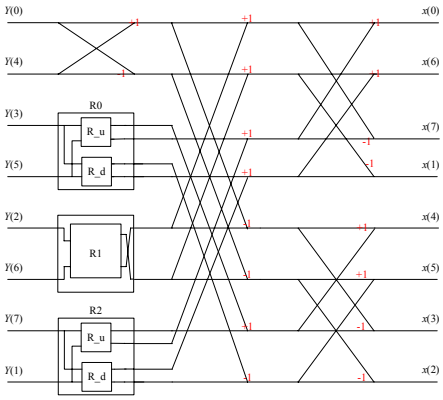

Fig. 2. Data flow of 8-point 1-D IDCT

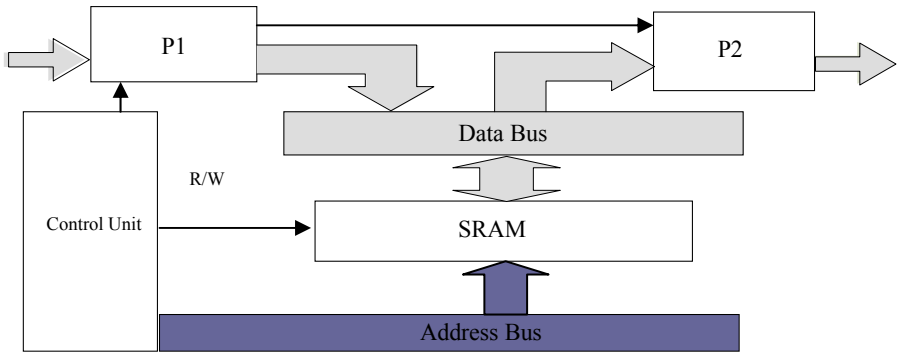

Fig. 3. The proposed architecture for 2-D DCT/IDCT processor.
(P1and P2: 1-D DCT/IDCT processor)

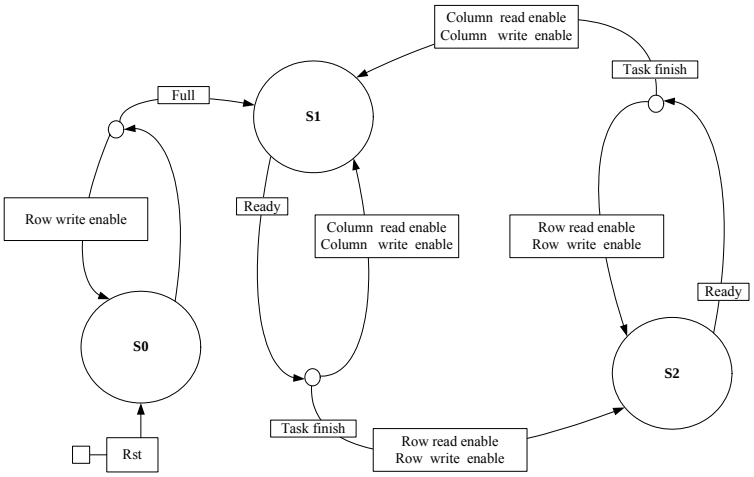

Fig. 4. The finite state machine of control unit

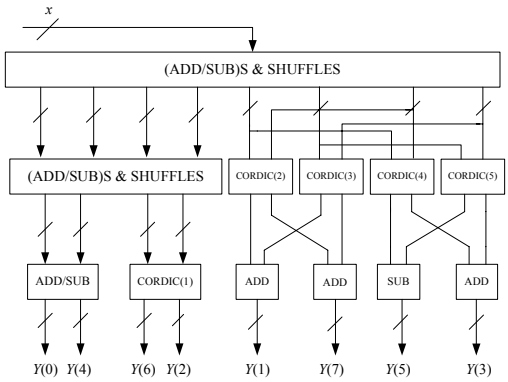

Fig. 5. The proposed 8-point DCT processor

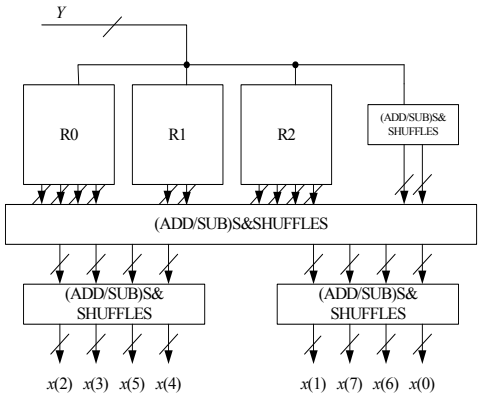

Fig. 6. The proposed 8-point IDCT processor

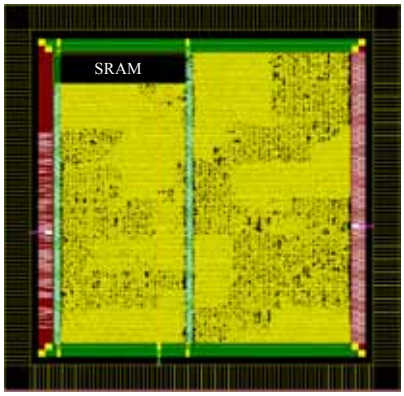

Fig.7. The layout view of the implemented 2-D DCT processor

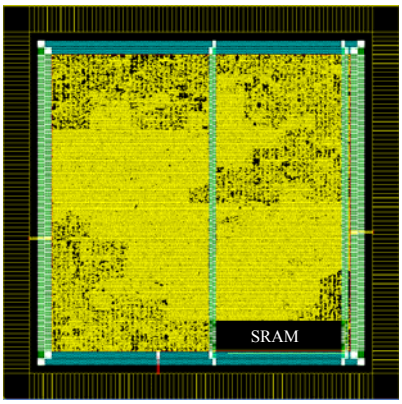

Fig.8. The layout view of the implemented 2-D IDCT processor