

A Novel Linear Array for Discrete Cosine Transform

Yaw-Shih Shieh* Tze-Yun Sung* Hsi-Chin Hsin**

*Department of Microelectronics Engineering
Chung Hua University
707, Sec. 2, Wufu Road, Hsinchu City 300-12
TAIWAN

**Department of Computer Science and Information Engineering
National United University
1, Lien-Da, Miaoli 360-03
TAIWAN

ysdaniel@chu.edu.tw, bobsung@chu.edu.tw, hsin@nuu.edu.tw

Abstract: - Discrete cosine transform (DCT) and inverse DCT (IDCT) have been widely used in many image processing systems. In this paper, a novel linear-array of DCT and IDCT is derived from the data flow of subband decompositions representing the factorized coefficient matrices in the matrix formulation of the recursive algorithm. For increasing the throughput as well as decreasing the hardware cost, the input and output data are reordered. The proposed 8-point DCT/IDCT processor with four multipliers, simple adders, and less registers and ROM storing the immediate results and coefficients, respectively, has been implemented on FPGA. The linear-array DCT/IDCT processor with the computation complexity $O(5N/8)$ and hardware complexity $O(N/2)$ is fully pipelined and scalable for variable length DCT/IDCT computations.

Key-Words: - DCT/IDCT, subband decomposition, linear-array, pipelined, scalable.

1 Introduction

Discrete cosine transform (DCT) is one of the major operations in various image/video compression standards [1]. Though fast Fourier transform (FFT) can be used to implement DCT, it requires complex-valued computations; and moreover, N -point DCT by FFT contains $O(\log 2N + 1)$ stages. The conventional DCT architectures using distributed arithmetic involve complex hardware with a great number of registers [2-6]. Other commonly used DCT architectures with matrix formulation and distributed memory [7-11] are however not suited for VLSI implementation because the hardware complex is proportional to the length of DCT, which leads to the scalability problem of variable length DCT computations. In this paper, we propose a novel linear-array architecture for scalable DCT/IDCT implementation.

The remainder of this paper proceeds as follows. In section 2, we propose the fast DCT/IDCT algorithm based on subband decomposition. In section 3, a programmable and reconfigurable FPGA-based implementation with low hardware cost is proposed for the fast DCT/IDCT computation. The performance comparison with conclusions can be found in section 4.

The National Science Council of Taiwan, under Grant NSC98-2221-E-216-037 supported this work.

Corresponding author: Tze-Yun Sung, bobsung@chu.edu.tw.

2 Fast DCT/IDCT Algorithm

For a N -point signal, $x[n]$, the discrete cosine transform (DCT) [1] is defined as

$$C[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left[\frac{(2n+1)k\pi}{2N}\right] \quad (1)$$

where $k = 0, \dots, N-1$, $\alpha[0] = 1/\sqrt{N}$, and $\alpha[k] = \sqrt{2/N}$ for $k > 0$. Let $x_L[n]$ and $x_H[n]$ denote the low-frequency and high-frequency subband signals of $x[n]$, respectively, which are defined as

$$x_L[n] = \frac{1}{2} \{x[2n] + x[2n+1]\} \quad (2)$$

$$x_H[n] = \frac{1}{2} \{x[2n] - x[2n+1]\} \quad (3)$$

where $n = 0, 1, 2, \dots, (N/2) - 1$.

As one can see, the DCT of $x[n]$ can be rewritten as

$$C[k] = \sum_{n=0}^{(N/2)-1} \alpha[k] x[2n] \cos\left(\frac{(4n+1)k\pi}{2N}\right) + \sum_{n=0}^{(N/2)-1} \alpha[k] x[2n+1] \cos\left(\frac{(4n+3)k\pi}{2N}\right)$$

$$= 2 \cos\left(\frac{\pi k}{2N}\right) \underbrace{\sum_{n=0}^{(N/2)-1} \alpha[k] x_L[n] \cos\left(\frac{(2n+1)k\pi}{N}\right)}_{C_L[k]} + 2 \sin\left(\frac{\pi k}{2N}\right) \underbrace{\sum_{n=0}^{(N/2)-1} \alpha[k] x_H[n] \sin\left(\frac{(2n+1)k\pi}{N}\right)}_{S_H[k]} \quad (4)$$

where $C_L[k]$ and $S_H[k]$ are the subband DCT and DST (discrete sine transform) of $x[n]$, respectively.

2.1 Fast DCT Algorithm Based on Subband Decomposition

Without loss of generality, the 8-point fast DCT algorithm based on subband decomposition is proposed for the widely used JPEG and MPEG-1/2 standards, which can be easily extended to variable length DCT computations. The vector form of 8-point DCT can be written as

$$\mathbf{C}_8 = [\mathbf{T}_{SB_DCT,8} \quad \mathbf{T}_{SB_DST,8}]_{8 \times 8} \cdot \begin{bmatrix} \mathbf{x}_L \\ \mathbf{x}_H \end{bmatrix}_{8 \times 1} \quad (5)$$

where $\mathbf{C}_8 = [C[0] \cdots C[7]]^T$, $\mathbf{x}_L = [x_L[0] \cdots x_L[3]]^T$, $\mathbf{x}_H = [x_H[0] \cdots x_H[3]]^T$, and $\mathbf{T}_{SB_DCT,8}$ and $\mathbf{T}_{SB_DST,8}$ denote the 8×4 matrices of subband DCT and subband DST, respectively, which can form orthonormal bases for the two orthogonal subspaces of \mathbf{R}^8 .

The data flow of computing the 2-point subband DCT: $\mathbf{C}_{LL,2}$ and subband DST: $\mathbf{C}_{LH,2}$ for the 8-point DCT is shown in Fig. 1. As one can see, data flow of computing $\mathbf{C}_{HL,2}$ and $\mathbf{C}_{HH,2}$ can be obtained in a similar way, and therefore is not shown in Fig. 1. All of the 2-point subband DCT and DST are given by

$$\mathbf{C}_{LL,2} = [\mathbf{T}_{SB_DCT,2} \quad \mathbf{T}_{SB_DST,2}]_{2 \times 2} \cdot \begin{bmatrix} \mathbf{x}_{LLL} \\ \mathbf{x}_{LLH} \end{bmatrix}_{2 \times 1} \quad (6)$$

$$\mathbf{C}_{LH,2} = [\mathbf{T}_{SB_DCT,2} \quad \mathbf{T}_{SB_DST,2}]_{2 \times 2} \cdot \begin{bmatrix} \mathbf{x}_{LHL} \\ \mathbf{x}_{LHH} \end{bmatrix}_{2 \times 1} \quad (7)$$

$$\mathbf{C}_{HL,2} = [\mathbf{T}_{SB_DCT,2} \quad \mathbf{T}_{SB_DST,2}]_{2 \times 2} \cdot \begin{bmatrix} \mathbf{x}_{HLL} \\ \mathbf{x}_{HLH} \end{bmatrix}_{2 \times 1} \quad (8)$$

$$\mathbf{C}_{HH,2} = [\mathbf{T}_{SB_DCT,2} \quad \mathbf{T}_{SB_DST,2}]_{2 \times 2} \cdot \begin{bmatrix} \mathbf{x}_{HHL} \\ \mathbf{x}_{HHH} \end{bmatrix}_{2 \times 1} \quad (9)$$

Thus, we have

$$\begin{bmatrix} \mathbf{C}_{LL,2} \\ \mathbf{C}_{LH,2} \\ \mathbf{C}_{HL,2} \\ \mathbf{C}_{HH,2} \end{bmatrix} = \mathbf{R}_8 \cdot \mathbf{x}_8 \quad (10)$$

where $\mathbf{x}_8 = [x[0] \cdots x[7]]^T$ is the original signal,

and

$$\mathbf{R}_8 = \frac{\sqrt{2}}{8} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (11)$$

Similarly, we have the following

$$\mathbf{C}_{L,4} = [\mathbf{T}_{SB_DCT,4} \quad \mathbf{T}_{SB_DST,4}]_{4 \times 4} \cdot \begin{bmatrix} \mathbf{x}_{LL,2} \\ \mathbf{x}_{LH,2} \end{bmatrix}_{4 \times 1} \quad (12)$$

$$\mathbf{C}_{H,4} = [\mathbf{T}_{SB_DCT,4} \quad \mathbf{T}_{SB_DST,4}]_{4 \times 4} \cdot \begin{bmatrix} \mathbf{x}_{HL,2} \\ \mathbf{x}_{HH,2} \end{bmatrix}_{4 \times 1} \quad (13)$$

Fig. 2 depicts the relationship between $\hat{\mathbf{C}}_{LL,4}$ and $\mathbf{C}_{LL,2}$, which can be obtained by the following:

$$\hat{\mathbf{C}}_{LL,4} = \mathbf{T}_{SB_DCT,4} \cdot \mathbf{x}_{LL,2} \quad (14)$$

$$\mathbf{C}_{LL,2} = \mathbf{T}_2 \cdot \mathbf{x}_{LL,2} \quad (15)$$

where \mathbf{T}_2 is the 2×2 transform matrix of the conventional 2-point DCT. Hence, equation (14) can be rewritten as

$$\hat{\mathbf{C}}_{LL,4} = \mathbf{T}_{SB_DCT,4} \cdot \mathbf{T}_2^{-1} \cdot \mathbf{C}_{LL,2} \quad (16)$$

The relationship between $\hat{\mathbf{S}}_{LH,4}$ and $\mathbf{C}_{LH,2}$ shown in Fig. 3 is based on the following.

$$\hat{\mathbf{S}}_{LH,4} = \mathbf{T}_{SB_DST,4} \cdot \mathbf{x}_{LH,2} \quad (17)$$

$$\mathbf{C}_{LH,2} = \mathbf{T}_2 \cdot \mathbf{x}_{LH,2} \quad (18)$$

Thus, we have

$$\hat{\mathbf{S}}_{LH,4} = \mathbf{T}_{SB_DST,4} \cdot \mathbf{T}_2^{-1} \cdot \mathbf{C}_{LH,2} \quad (19)$$

Similarly, based on equation (5) and the following equations:

$$\mathbf{C}_{L,4} = \mathbf{T}_4 \cdot \mathbf{x}_{L,4} \quad (20)$$

$$\mathbf{C}_{H,4} = \mathbf{T}_4 \cdot \mathbf{x}_{H,4} \quad (21)$$

where \mathbf{T}_4 is the 4×4 transform matrix of the conventional 4-point DCT. We have

$$\hat{\mathbf{C}}_{L,8} = \mathbf{T}_{SB_DCT,8} \cdot \mathbf{x}_{L,4} = \mathbf{T}_{SB_DCT,8} \cdot \mathbf{T}_4^{-1} \cdot \mathbf{C}_{L,4} \quad (22)$$

$$\hat{\mathbf{C}}_{H,8} = \mathbf{T}_{SB_DST,8} \cdot \mathbf{x}_{H,4} = \mathbf{T}_{SB_DST,8} \cdot \mathbf{T}_4^{-1} \cdot \mathbf{C}_{H,4} \quad (23)$$

Fig.4 depicts data flow of computing $\mathbf{C}_{L,4}$ and $\mathbf{C}_{H,4}$ using 4-point subband DCT and DST. Fig. 5

depicts data flow of computing $\hat{C}_{L,8}$ and $C_{L,4}$ based on subband decomposition. Data flow of computing $\hat{S}_{H,8}$ and $C_{H,4}$ based on subband decomposition is shown in Fig. 6. Data flow of computing C_8 using 8-point subband DCT and DST is shown in Fig. 7. In other words, C_8 can be obtained by

$$C_8 = \hat{C}_{L,8} + \hat{S}_{H,8} \quad (24)$$

Base on eqs. (12), (13), (16), (19), (22) and (23), we have

$$C_8 = F_8 \cdot [C_{LL,2}^T \quad C_{LH,2}^T \quad C_{HL,2}^T \quad C_{HH,2}^T]^T \quad (25)$$

where

$$F_8 = [K_3 \quad K_4]_{8 \times 8} \cdot \begin{bmatrix} [K_1 \quad K_2]_{4 \times 4} & 0 \\ 0 & [K_1 \quad K_2]_{4 \times 4} \end{bmatrix}_{8 \times 8} \quad (26)$$

$$K_1 = \begin{bmatrix} 1.4142 & 0 \\ 0 & 1.3066 \\ 0 & 0 \\ 0 & -0.5412 \end{bmatrix} \quad (27)$$

$$K_2 = \begin{bmatrix} 0 & 0 \\ 0.5412 & 0 \\ 0 & 1.4142 \\ 1.3066 & 0 \end{bmatrix} \quad (28)$$

$$K_3 = \begin{bmatrix} 1.412 & 0 & 0 & 0 \\ 0 & 1.3870 & 0 & 0 \\ 0 & 0 & 1.3066 & 0 \\ 0 & 0 & 0 & 1.1759 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.7857 \\ 0 & 0 & -0.5412 & 0 \\ 0 & -0.2759 & 0 & 0 \end{bmatrix} \quad (29)$$

$$K_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2549 & 0 & -0.1056 & 0 \\ 0 & 0.5 & 0 & -0.2071 \\ 0.3007 & 0 & 0.7259 & 0 \\ 0 & 0.5412 & 0 & 1.3066 \\ 0.4500 & 0 & 1.0864 & 0 \\ 0 & 1.2071 & 0 & -0.5 \\ 1.2815 & 0 & -0.5308 & 0 \end{bmatrix} \quad (30)$$

According to equations (27) ~ (30), we have

$$F_8 = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.8123 & 0.7507 & 0 & 0.3605 & 0 & 0 & -0.1493 \\ 0 & 0 & 0 & 1.8478 & 0 & 0.7654 & 0 & 0 \\ 0 & -0.6364 & 1.5364 & 0 & 0.4252 & 0 & 0 & 1.0266 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0.4252 & -1.0266 & 0 & 0.6364 & 0 & 0 & 1.5364 \\ 0 & 0 & 0 & -0.7654 & 0 & 1.8478 & 0 & 0 \\ 0 & -0.3605 & -0.1493 & 0 & 1.8123 & 0 & 0 & -0.7507 \end{bmatrix} \quad (31)$$

Finally, the proposed 8-point DCT computation based on subband decomposition is as follows:

$$C_8 = \hat{F}_8 \cdot R_8 \cdot x_8 \quad (32)$$

where

$$\hat{F}_8 = 2 \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9239 & 0.3827 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.3827 & 0.9239 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9062 & 0.3754 & 0.1802 & -0.0746 \\ 0 & 0 & 0 & 0 & -0.1802 & -0.0746 & 0.9062 & -0.3754 \\ 0 & 0 & 0 & 0 & -0.3182 & 0.7682 & 0.2126 & 0.5133 \\ 0 & 0 & 0 & 0 & 0.2126 & -0.5133 & 0.3182 & 0.7682 \end{bmatrix} \quad (33)$$

Fig. 8 shows block diagram of the proposed DCT computation; one of the advantages is that R_8 is orthogonal, and all of the sub-matrices of \hat{F}_8 are orthonormal.

2.2 Fast IDCT Algorithm Based on Subband Decomposition

According to eq. (31), IDCT can be obtained by

$$x_8 = R_8^{-1} \cdot \hat{F}_8^{-1} \cdot C_8 \quad (34)$$

where

$$R_8^{-1} = \frac{8}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (35)$$

$$\hat{F}_8^{-1} = \frac{1}{2} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9239 & -0.3827 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3827 & 0.9239 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.9062 & -0.1802 & -0.3182 & 0.2126 \\ 0 & 0 & 0 & 0 & 0.3754 & 0.3754 & 0.7682 & -0.5133 \\ 0 & 0 & 0 & 0 & 0.1802 & 0.1802 & 0.2126 & 0.3182 \\ 0 & 0 & 0 & 0 & -0.0746 & -0.0746 & 0.5133 & 0.7682 \end{bmatrix} \quad (36)$$

As \mathbf{R}_8 is orthogonal and all of the sub-matrices of $\hat{\mathbf{F}}_8$ are orthonormal, the inverse of \mathbf{R}_8 and $\hat{\mathbf{F}}_8$ can be obtained easily. In addition, it takes only twenty multiplication operations for both DCT and IDCT.

3 A Linear Array for DCT and IDCT

Based on the proposed approach to fast DCT computation shown in Fig. 8, an efficient architecture for implementing the fast DCT/IDCT processor is thus presented in this section. Recall that the DCT of a signal, \mathbf{x}_8 , can be efficiently obtained by $\mathbf{C}_8 = \hat{\mathbf{F}}_8 \cdot \mathbf{R}_8 \cdot \mathbf{x}_8$. Let $\mathbf{y}_8 = \mathbf{R}_8 \cdot \mathbf{x}_8$, then we have $\mathbf{C}_8 = \hat{\mathbf{F}}_8 \cdot \mathbf{y}_8$. The matrix-vector multiplication of $\mathbf{R}_8 \cdot \mathbf{x}_8$, in which six CSA(3,2)s (carry-save-adder (3,2)) and one CLA (carry-look-ahead-adder) [12-13] are utilized, and therefore four simple-addition time and one CLA computation time is required to compute each element of \mathbf{y}_8 . The multiplier-array (MA) consisted of four multipliers and the CLA-array (CA) consisted of eight CLAs, respectively, which are used to compute the matrix-vector computation of $\hat{\mathbf{F}}_8 \cdot \mathbf{y}_8$; thus, only one multiplication time with one CLA computation time is needed to compute each element of \mathbf{C}_8 , i.e. the DCT coefficient. Fig. 9 depicts data flow of the proposed fast DCT processor with pipelined linear-array architecture [14]. As a result, only five multiplication cycles with five addition cycles are needed to compute 8-point DCT. In general, for N -point DCT, the computation time and hardware complexity of the proposed fast DCT processor are $O(5N/8)$ and $O(N/2)$, respectively.

Figure 10 shows data flow of the proposed fast IDCT algorithm [14], where \mathbf{C}_8 is the DCT of an 8-point signal \mathbf{x}_8 ; $\mathbf{z}_8 = \hat{\mathbf{F}}_8^{-1} \cdot \mathbf{C}_8$, and $\mathbf{x}_8 = \mathbf{R}_8^{-1} \cdot \mathbf{z}_8$. The so-called full-CSA(4,2) (FCSA(4,2)) consisted of two CSA(3,2) and one CLA for the computation of \mathbf{z}_8 . It is noted that the CLA-array consisted of eight CLAs can also be used for the computation of \mathbf{x}_8 . As shown in Fig. 10, only five multiplication cycles with three addition cycles are needed to compute 8-point IDCT. As one can see, the computation time and hardware complexity of the proposed fast IDCT architecture are the same as that of the proposed fast DCT architecture. In addition, only 16-word RAM/registers and 10-word ROM are required to store the intermediate results and constants, respectively; and the latency time is only

5-multiplication-cycle.

Fig. 11 shows system block diagram of the proposed fast DCT/IDCT architecture. The platform for architecture development and verification has been designed as well as implemented in order to evaluate the development cost. Fig. 12 depicts block diagram of the platform. In which, the 8051 microcontroller reads data from PC via DMA channel and writes the result back to PC by USB 2.0 bus; the Xilinx[®] XC2V6000 FPGA chip implements the proposed DCT processor [15]. The architecture development and verification board shown in Fig. 13 is to verify and evaluate the proposed DCT/IDCT architecture. Moreover, the reusable intellectual property (IP) DCT/IDCT core has also been implemented in Matlab[®] for functional simulations. The hardware code written in Verilog[®] is running on a workstation with the ModelSim[®] simulation tool and Xilinx[®] ISE smart compiler. In addition, the FPGA platform shown in Fig. 12 is to verify and evaluate the proposed DCT architecture. It is noted that the throughput can be improved by using the proposed architecture while the computation accuracy is the same as that obtained by using the conventional one with the same word length. Thus, the proposed programmable DCT/IDCT architecture is able to improve the power consumption and computation speed significantly. The proposed DCT/IDCT processor used to compute 8/16/32/64-point DCT/IDCT are composed mainly of the 8-point DCT/IDCT core; the computation complexity using a single 8-point DCT/IDCT core is $O(5N/8)$ for extending N -point DCT/IDCT computation.

4 Conclusion

By taking advantage of subband decomposition, a high-efficiency architecture with pipelined structures is proposed for fast DCT/IDCT computation. Specifically, the proposed DCT/IDCT architecture not only improves throughput by more than two times that of the conventional architectures [2-6], but also saves memory space significantly [1]. Table 1 shows comparisons between the proposed architecture and the conventional architectures [2-6]. Table 2 shows comparisons with other commonly used architectures [1], [7-8]. In addition, the proposed fast DCT/IDCT architecture is highly regular, scalable, and flexible. The DCT/IDCT processor designed by using the portable and reusable Verilog[®] is a reusable IP, which can be implemented in various processes; combined with efficient use of hardware resources for trade-offs of

performance, area and power consumption; and therefore is much suited to the JPEG and MPEG-1/2 applications.

References:

- [1] T. Y. Sung, "Memory-Efficient and high-performance 2-D DCT and IDCT processors based on CORDIC rotation", *WSEAS Trans. Electronics*, Vol. 3, No. 12, Dec. 2006, pp.565-574.
- [2] Y. H. Hu, Z. Wu, "An efficient CORDIC array structure for the implementation of discrete cosine transform", *IEEE Trans. Signal Processing*, No. 1, Vol. 43, Jan. 1995, pp.331-336.
- [3] H. Jeong, J. Kim, W. K. Cho, "Low-power multiplierless DCT architecture using image data correlation", *IEEE Trans. Consumer Electronics*, No.1, Vol. 50, 2004, pp.262-267.
- [4] D. Gong, Y. He, Z. Gao, "New cost-effective VLSI implementation of a 2-discrete cosine Transform and its inverse", *IEEE Trans. Circuits Syst. for Video Technology*, Vol. 14, No. 4, April 2004, pp. 405-415.
- [5] V. Dimitrov, K. Wahid, G. Jullien, "Multiplication-free 8×8 2D DCT architecture using Algebraic integer encoding", *Electronics Letters*, No. 20, Vol. 40, 2004, pp.1310-1311.
- [6] M. Alam, W. Badawy, G. Jullien, "A new time distributed DCT architecture for MPEG-4 hardware reference model", *IEEE Trans. Circuits Syst. for Video Technology*, Vol.15, No.5, 2005, pp.726-730.
- [7] 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, 2002, pp.97-107.
- [8] H. S. Hou, "A fast recursive algorithm for computing the discrete cosine transform", *IEEE Trans. Acoust., Speech, Signal Processing*, No.10, Vol. ASSP-35, Oct. 1987, pp.1455-1461.
- [9] H. Elsimary, "A Reconfigurable VLSI Architecture for Low Power IDCT", *WSEAS Trans. Systems*, Vol. 5, No. 8, 2006, pp.1797-1801.
- [10] M. R. Bodnar, J. P. Durbano, J. R. Humphrey, P. F. Curt, D. W. Prather, "FPGA-Based, Floating-Point Reduction Operation", *10th WSEAS Int. Conf. Applied Mathematics*, 2006, pp.5-9.
- [11] T. Y. Sung, H. C. Hsin, "Reconfigurable VLSI architecture for FFT processor" *WSEAS Trans. Circuits and Systems*, Vol. 8, No. 6, 2009, pp.465-474.
- [12] I. Koren, *Computer arithmetic algorithm*, Second edition, A. K. Peters, Natick, MA, 2002, Chapter 5.
- [13] T. Y. Sung, H. C. Hsin, "Design and simulation of reusable IP CORDIC core for special-purpose processors," *IET Proc. Computers & Digital Techniques*, Vol. 1, No. 5, 2007, pp.581-589.
- [14] G. H. Golub, C. F. Van Loan, *Matrix computations*, The John Hopkins University Press, 1996, Chapter 6. Parallel matrix computations, pp.275-307.
- [15] Xilinx FPGA: <http://www.xilinx.com/products>.

Table 1 Comparisons of the proposed architecture and the conventional architectures

8-point DCT/IDCT	The conventional architectures	The proposed high- efficient architecture
	The parallel architectures with single memory-bank [2]-[6]	This work [Shieh, Sung, Hsin, 2010]
Processors	8	--
Real-multipliers	16	4
Real-Adders	18	26
RAM (Registers)	64	16
ROM	6	10
Hardware complexity	$O(N - \log_2 N + 1)$	$O(N/2)$
Computation complexity	$O(2N)$	$O(5N/8)$
Latency	16	5
Pipelability	no	yes
Scalability	poor	better
Power consumption	poor	better

Table 2 Comparisons of the proposed architecture and other commonly used architectures

8-point DCT/IDCT	Hsiao [7]	Hsiao [8]	Shieh, Sung, Hsin, 2010
Real-multipliers	-	-	4
CORDIC	-	3	-
Real-adders	10	14	26
Complex-Multipliers	3		-
Delay elements (Words)	171	-	-
Hardware complexity	$O(\log N)$	$O(\log N)$	$O(N/2)$
Computation complexity	$O(N \log N)$	$O(N \log N)$	$O(5N/8)$
Pipelability	no	yes	yes
Scalability	good	good	better

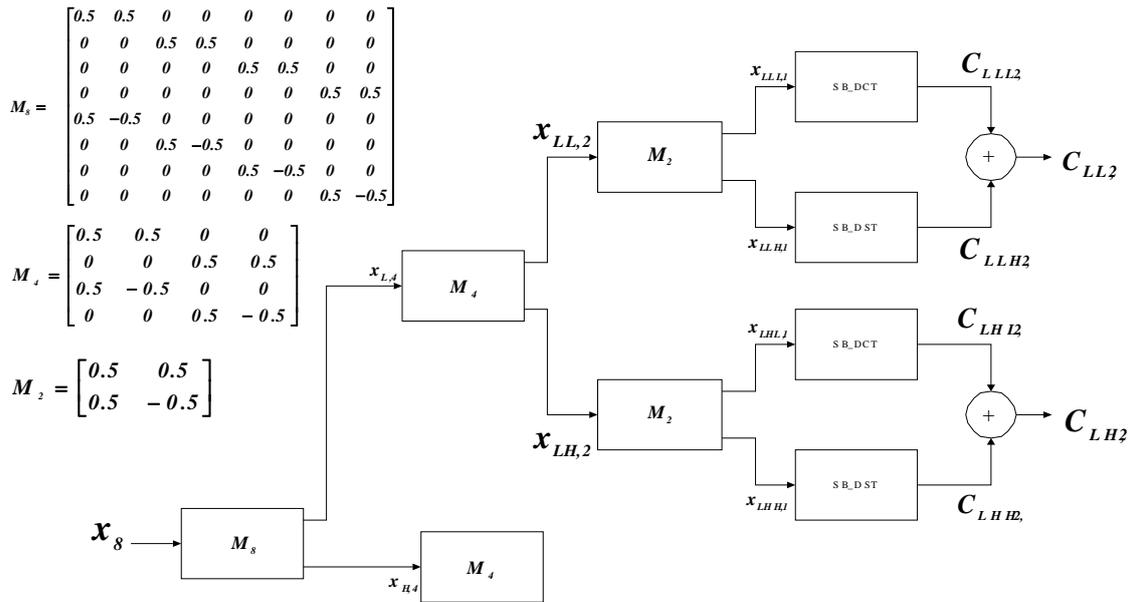


Fig. 1 Data flow of computing the 2-point subband DCT

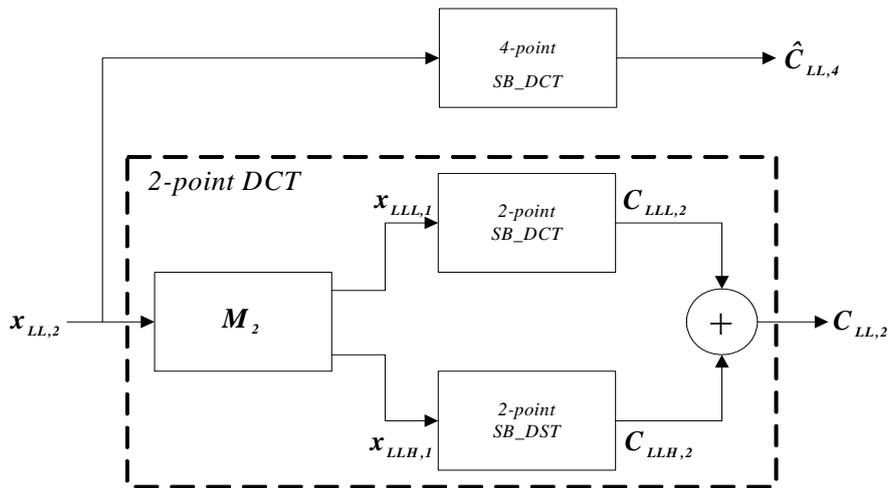


Fig. 2 Data flow of computing $\hat{C}_{LL,4}$ and $C_{LL,2}$ based on subband decomposition

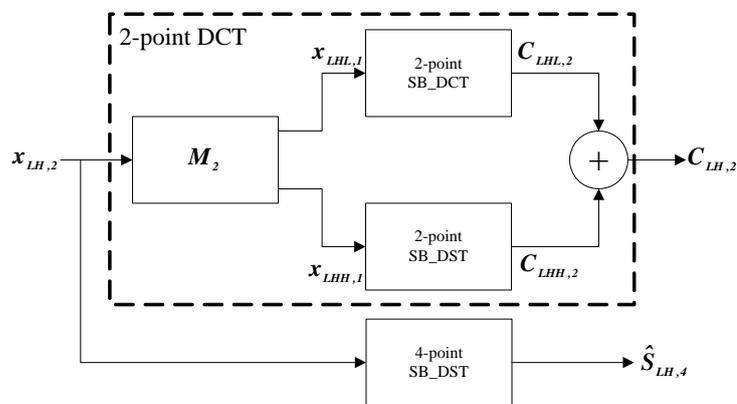


Fig. 3 Data flow of computing $C_{LH,2}$ and $\hat{S}_{LH,4}$ based on subband decomposition

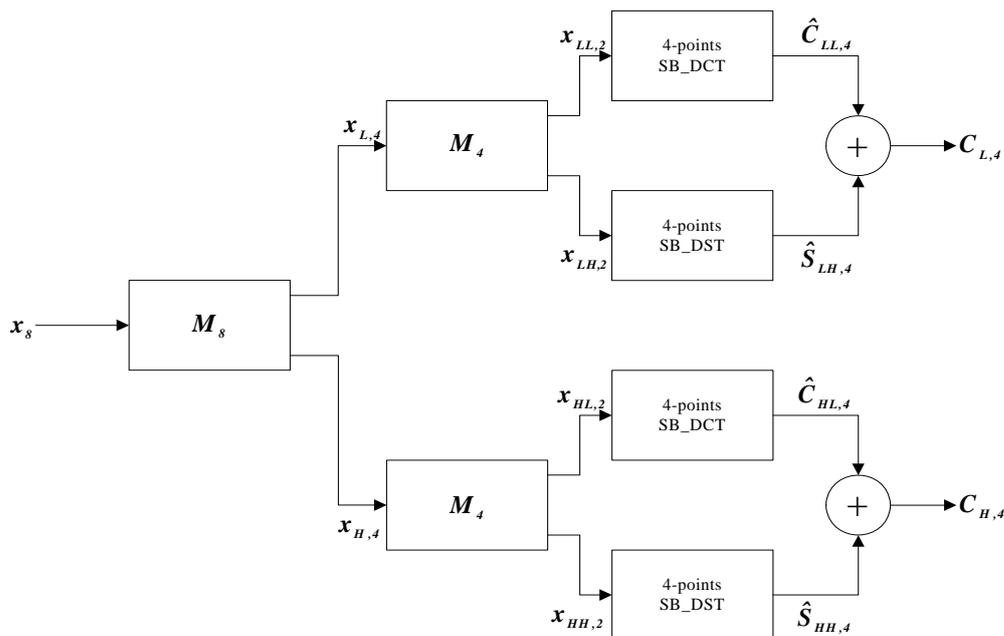


Fig. 4 Data flow of computing $C_{L,4}$ and $C_{H,4}$ using 4-point subband DCT and DST

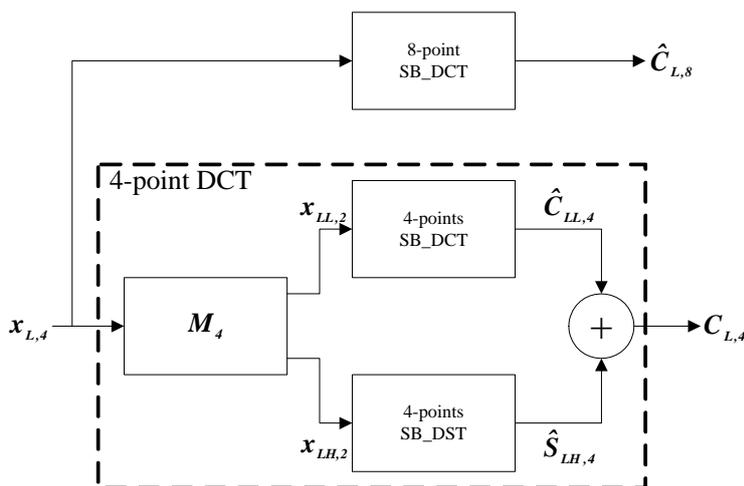


Fig. 5 Data flow of computing $\hat{C}_{L,8}$ and $C_{L,4}$ based on subband decomposition

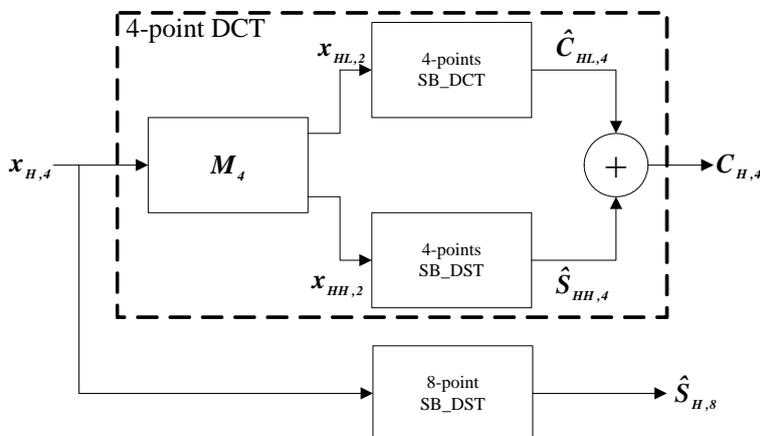


Fig. 6 Data flow of computing $\hat{S}_{H,8}$ and $C_{H,4}$ based on subband decomposition

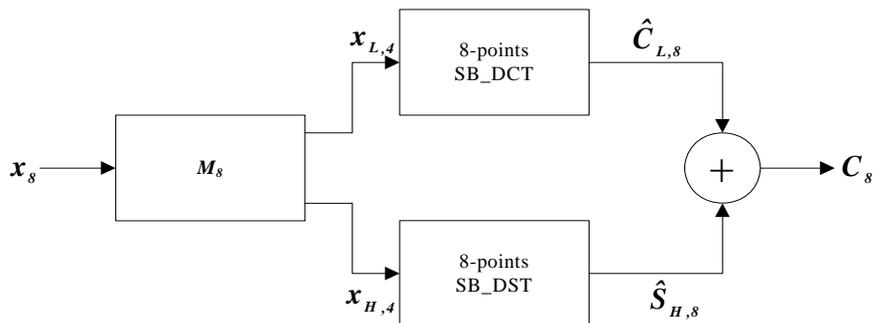


Fig. 7 Data flow of computing C_8 using 8-point subband DCT and DST

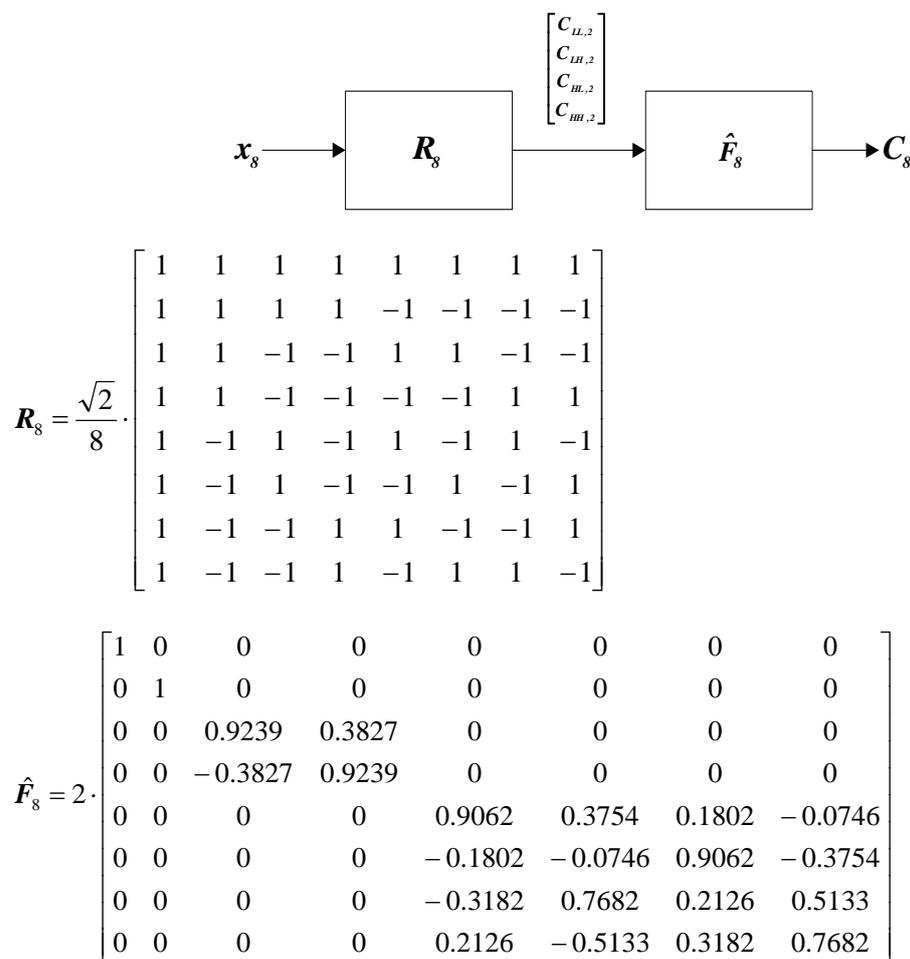


Fig. 8 Block diagram of the proposed (8-point) fast DCT algorithm based on subband decomposition

Cycles	FA	MA	CA
Add. _1	y[0]	--	C[0]
Add. _2	y[1]	--	C[1]
Add. _3	y[2]	--	--
Mul. _1	y[3]	$y[2] \cdot 0.9239, y[2] \cdot (-0.3827)$ $y[3] \cdot 0.3827, y[3] \cdot (0.9239)$	--
Add. _4	y[4]	--	C[2], C[3]
Mul. _2	y[5]	$y[4] \cdot 0.9062, y[4] \cdot (-0.1802),$ $y[4] \cdot (-0.3182), y[4] \cdot 0.2126$	--
Mul. _3	y[6]	$y[5] \cdot 0.3754, y[5] \cdot (-0.0746),$ $y[5] \cdot 0.7682, y[5] \cdot 0.5133$	--
Mul. _4	y[7]	$y[6] \cdot 0.1802, y[6] \cdot 0.9062,$ $y[6] \cdot 0.2126, y[6] \cdot 0.3182$	--
Mul. _5	----	$y[7] \cdot (-0.0746), y[7] \cdot (-0.3754),$ $y[7] \cdot 0.5133, y[7] \cdot 0.7682$	--
Add. _5	----		C[4], C[5], C[6], C[7]

Fig.9 Data flow of the proposed fast DCT processor with pipelined linear-array architecture (Add. _: addition-cycle, Mul._: multiplication-cycle)

Cycles	MA	FCSA	CA
Mul. _1	$C[2] \cdot 0.9239, C[3] \cdot (-0.3827)$ $C[2] \cdot 0.3827, C[3] \cdot 0.92393$	z[0], z[1]	--
Mul. _2	$C[4] \cdot 0.9062, C[5] \cdot (-0.1802),$ $C[6] \cdot (-0.3182), C[7] \cdot 0.2126$	z[2], z[3]	$C_{0}+C_{1}=C_{01}$
Mul. _3	$C[4] \cdot 0.3754, C[5] \cdot 0.3754,$ $C[6] \cdot 0.7682, C[7] \cdot (-0.5133)$	z[4]	$C_{01}+C_{2}=C_{02}$
Mul. _4	$C[4] \cdot (-0.3182), C[5] \cdot 0.7682,$ $C[6] \cdot 0.2126, C[7] \cdot 0.5144$	z[5]	$C_{02}+C_{3}=C_{03}$
Mul. _5	$C[4] \cdot 0.2126, C[5] \cdot (-0.5133),$ $C[6] \cdot 0.3182, C[7] \cdot 0.7682$	z[6]	$C_{03}+C_{4}=C_{04}$
Add. _1	--	z[7]	$C_{04}+C_{5}=C_{05}$
Add. _2	--	--	$C_{05}+C_{6}=C_{06}$
Add. _3	--	--	$C_{06}+C_{7}=C_{07}$ x[0], x[1], x[2], x[3], x[4], x[5], x[6], x[7]

Fig. 10 Data flow of the proposed fast IDCT processor with pipelined linear-array architecture

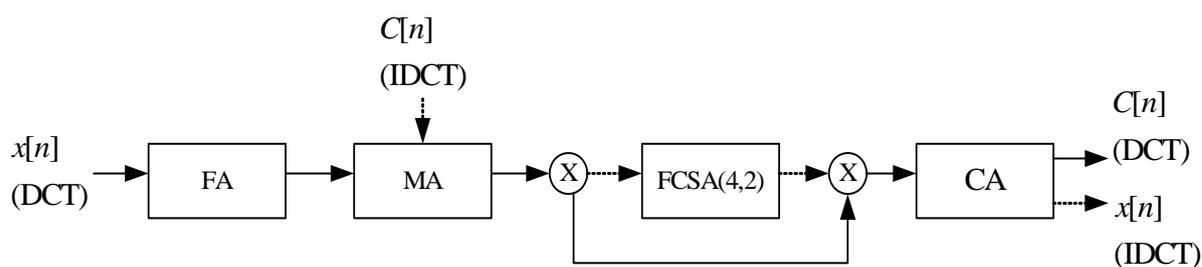


Fig. 11 System block diagram of the proposed DCT/IDCT architecture

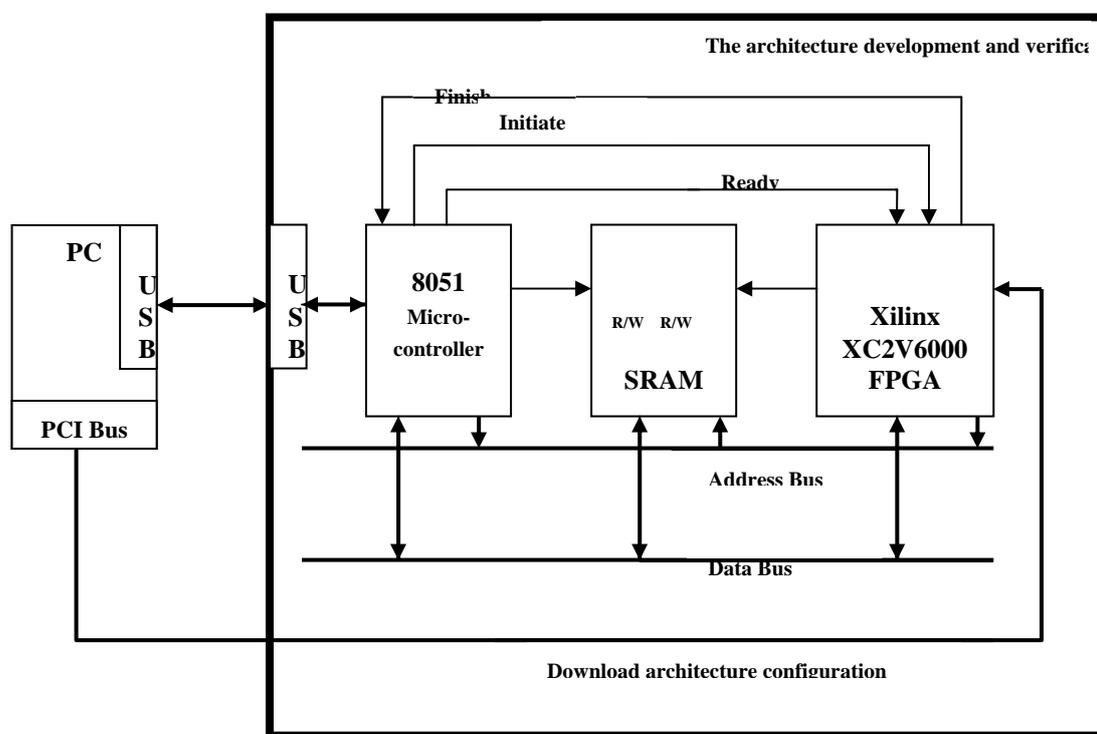


Fig. 12 Block diagram of the architecture development and verification platform for the proposed DCT/IDCT processor.

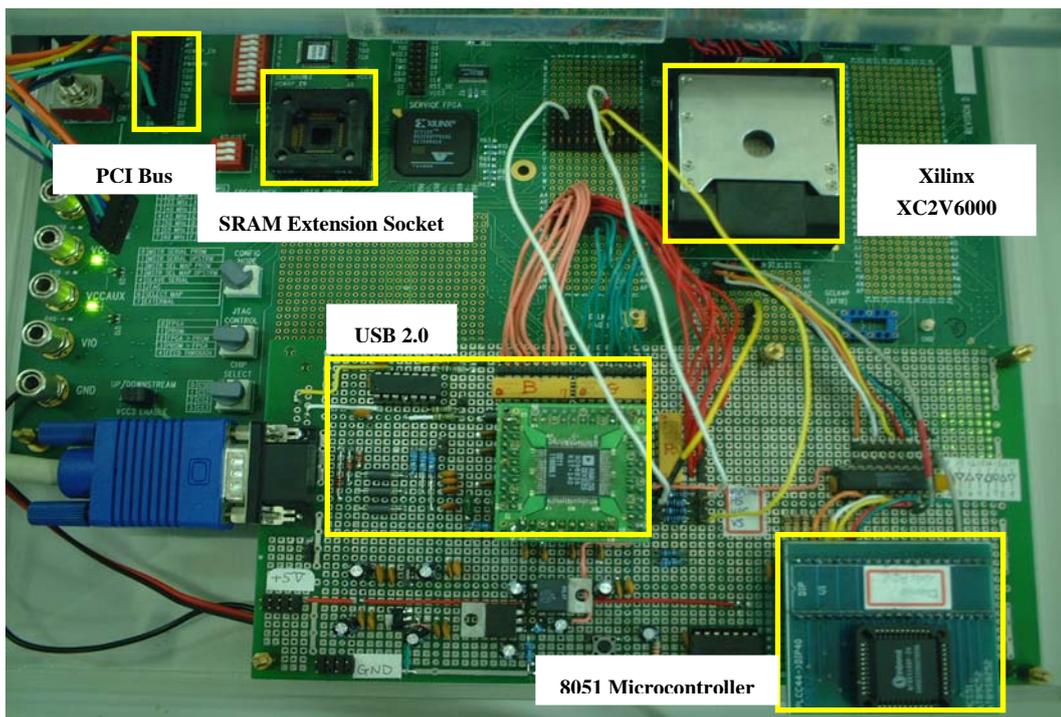


Fig. 13 The architecture development and verification board.