

行政院國家科學委員會專題研究計畫 成果報告

快速座標旋轉原理演算法應用於特殊數位信號處理之晶片 設計與製作 研究成果報告(精簡版)

計畫類別：個別型
計畫編號：NSC 97-2221-E-216-044-
執行期間：97年08月01日至98年07月31日
執行單位：中華大學微電子工程學系

計畫主持人：宋志雲
共同主持人：辛錫進
計畫參與人員：碩士班研究生-兼任助理人員：柯律廷

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中華民國 98年07月20日

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

快速座標旋轉原理演算法應用於特殊數位信號處理之晶片設計與製作

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC 97-2221-E-216-044-

執行期間：97年8月1日至98年7月31日

計畫主持人：宋志雲

共同主持人：辛錫進

計畫參與人員：柯律廷

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：中華大學

中華民國 98 年 7 月 20 日

中文摘要

本研究報告第一部分說明一個座標旋轉原理(CORDIC)為基礎之分離式基底快速傅利葉轉換(split-radix fast Fourier transform)核心用於正交頻分複用技術(OFDM)，例如超寬頻無線網絡(UWB)，非對稱數位用戶線路(ADSL)，數位音訊廣播(DAB)，數位視訊廣播－陸上系統(DVB-T)，高速數位用戶線路(VHDSL)以及全球互通微波存取(WiMAX)。高速128/256/512/1024/2048/4096/8192點快速傅利葉轉換處理器於本研究中以台積電製程0.18微米(1p6m)完成晶片設計，所有控制電路均集成於單一晶片上。所完成之快速傅利葉轉換處理器在功率消耗上以及晶片面積上，均獲得較佳之效率。

關鍵詞-矽智產權(Intellectual Property)，快速傅利葉轉換，分離式基底，座標旋轉原理，正交頻分複用技術。

本研究報告第二部分說明一個超大型積體電路架構之二維順向及反向 8×8 正弦轉換處理器，此一處理器的架構具備平行及疊流結構，其中包含64位元之靜態隨機記憶體，6個位元的唯讀記憶體置放參數，與傳統演算法之架構相比較，節省記憶體空間。此一處理器中的乘法器悉數被座標旋轉原理處理器取代，節省許多硬體，同時減少功率消耗，增加多效率。

關鍵詞-順向及反向正弦轉換，平行及疊流結構，低功率，座標旋轉原理。

英文摘要

The first part of report presents a Coordinate Rotation Digital Computer (CORDIC)-based split-radix fast Fourier transform (FFT) core for Orthogonal Frequency Division Multiplexing (OFDM) systems, for example, Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL), and Worldwide Interoperability for Microwave Access (WiMAX). High-speed 128/256/512/1024/ 2048/4096/8192-point FFT processors have been implemented by 0.18 μm (1p6m) at 1.8V, in which all the control signals are generated internally. These FFT processors outperform the conventional ones in terms of both power consumption and core area.

Key-Words: - Intellectual Property (IP), FFT, split-radix, CORDIC, OFDM.

Two-dimensional discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) have been widely used in many image processing systems. In the second part of report, 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.

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

報告內容：

一、可重組式快速傅利葉轉換之超大型積體電路架構

1 Introduction

High-performance fast Fourier transform (FFT) processor is needed especially for real-time digital signal processing applications. Specifically, the computation of discrete Fourier transform (DFT) ranging from 128 to 8192 points is required for the orthogonal frequency division multiplexer (OFDM) of the following standards: Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL) and Worldwide Interoperability for Microwave Access (WiMAX) [1]-[8]. Thompson [9] proposed an efficient VLSI architecture for FFT in 1983. Wold and Despain [10] proposed pipelined and parallel-pipelined FFT for VLSI implementations in 1984. Widhe [11] developed efficient processing elements of FFT in 1997. To reduce the computation complexity, the split-radix 2/4, 2/8, and 2/16 FFT algorithms were proposed in [12]-[15].

As the Booth multiplier is not suitable for hardware implementations of large FFT, we propose the CORDIC-based multiplier. Moreover, we develop a ROM-free twiddle factor generator using simple shifters and adders only [1], which obviates the need to store all the twiddle factors in a large ROM space. As a result, the proposed CORDIC-based split-radix FFT core with the ROM-free twiddle factor generator is suitable for the wireless local area network (WLAN) applications. In this report, a high-performance 128/256/512/1024/2048/4096/8192-point FFT processor is presented for the European and Japanese standards. The remainder of this report proceeds as follows. In Section 2, the split-radix 2/8 FFT algorithm and the CORDIC algorithm are reviewed briefly. In Section 3, the reusable IP 128-point CORDIC-based split-radix FFT core is proposed. In Section 4, the hardware implementations of FFT processors are described. The performance analysis is presented in Section 5. Finally, the conclusion is given in Section 6.

2 Review of Split-Radix FFT and CORDIC Algorithm

2.1 Split-Radix FFT

The idea behind the split-radix FFT algorithm is to compute the even and odd terms of FFT separately. The even term of the split-radix 2/8 FFT algorithm is given by

$$X(2k) = \sum_{n=0}^{N/2-1} (x(n) + x(n + \frac{N}{2}))W_{N/2}^{nk} \quad (1)$$

where $W_{N/2} = e^{-j\frac{2\pi}{N/2}}$ and $k = 0, 1, 2, \dots, (N/2) - 1$. The odd term is as follows:

$$\begin{aligned} X(8k+l) = & \sum_{n=0}^{N/8-1} ((x(n) + x(n + \frac{2N}{8})W_4^l + x(n + \frac{4N}{8})W_4^{2l} \\ & + x(n + \frac{6N}{8})W_4^{-l}) + (x(n + \frac{N}{8}) + x(n + \frac{3N}{8})W_4^l \\ & + x(n + \frac{5N}{8})W_4^{2l} + x(n + \frac{7N}{8})W_4^{-l})W_N^{nl}W_{N/8}^{nk} \end{aligned} \quad (2)$$

where $k = 0, 1, 2, \dots, (N/8) - 1$ and $l = 1, 3, 5, 7$. The split-radix 2/8 FFT algorithm, which

combined with radix-2 and radix-4 proves effective to develop a reusable IP 128-point FFT core.

2.2 CORDIC Algorithm

The CORDIC algorithm in the circular coordinate system is as follows [16].

$$x(i+1) = x(i) - \sigma_i 2^{-i} y(i) \quad (3)$$

$$y(i+1) = y(i) + \sigma_i 2^{-j} x(i) \quad (4)$$

$$z(i+1) = z(i) - \sigma_i \alpha(i) \quad (5)$$

$$\alpha(i) = \tan^{-1} 2^{-i} \quad (6)$$

where $\sigma_i = \text{sign}(z(i))$ with $z(i) \rightarrow 0$ in the rotation mode, and $\sigma_i = -\text{sign}(x(i)) \cdot \text{sign}(y(i))$

with $y(i) \rightarrow 0$ in the vectoring mode. The scale factor: $k(i)$ is equal to $\sqrt{1 + \sigma_i^2 2^{-2i}}$. After n

micro-rotations, the product of the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k(i) = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \quad (7)$$

Notice that CORDIC in the circular coordinate system with rotation mode can be written by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = K_c \begin{bmatrix} \cos z_0 & \sin z_0 \\ -\sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (8)$$

where $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ and $\begin{bmatrix} x_n \\ y_n \end{bmatrix}$ are the input vector and the output vector, respectively, z_0 is the

rotation angle, and K_c is the scale factor. In [1], the circular rotation computation of CORDIC was used for complex multiplication with $e^{-j\theta}$, which is given by

$$\begin{bmatrix} \text{Re}[X'] \\ \text{Im}[X'] \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \text{Re}[X] \\ \text{Im}[X] \end{bmatrix} \quad (9)$$

3 Reusable IP 128-Point CORDIC-Based Split-Radix FFT Core

Figure 1 shows the proposed 128-point CORDIC-based split-radix FFT processor, which can be used as a reusable IP core for various FFT with multiples of 128 points. Notice that the modified split-radix 2/8 FFT butterfly processor and the ROM-free twiddle factor generator are used. In addition, an internal (128 32-bit) SRAM is used to store the input and output data for hardware efficiency, through the use of the in-place computation algorithm [1].

3.1 CORDIC-Based Split-Radix 2/8 FFT Processor

For the butterfly computation of the proposed CORDIC-based split-radix 2/8 FFT processor, sixteen complex additions, two constant multiplications (CM), and four CORDIC operations are needed, as shown in Figure 2. The CORDIC algorithm has been widely used in various DSP applications because of the hardware simplicity. According to equation (9), the twiddle factor multiplication of FFT can be considered a 2-D vector rotation in the circular coordinate system. Thus, CORDIC in the circular coordinate system with rotation mode is adopted to compute complex multiplications of FFT.

The pipelined CORDIC arithmetic unit can be obtained by decomposing the CORDIC algorithm into a sequence of operational stages. In [17], we derived the error analysis of fixed-point CORDIC arithmetic, based on which, the number of the CORDIC stages can be determined effectively. For example, the number of the CORDIC stages is 12 if the overall relative error of 16-bit CORDIC arithmetic is required to be less than 10^{-3} . The pipelined CORDIC arithmetic unit with 12 stages and an additional pre-scalar stage. In which, the pre-calculated scaling factor $K_c \approx 1.64676$ and the Booth binary recoded format leads to 1.101001. The main concern for the design of the CORDIC arithmetic unit is throughput rather than latency. The proposed CORDIC arithmetic unit in terms of gate counts is less than 4 real multipliers significantly. In addition, the power consumption can be reduced significantly by using the proposed CORDIC arithmetic unit; it has been reduced by 30% according to the report of PrimePower® distributed by Synopsys.

As the twiddle factors: W_8^1 and W_8^3 are equal to $\frac{\sqrt{2}}{2}(1-j)$ and $-\frac{\sqrt{2}}{2}(1+j)$, respectively, a complex number, say $(a+bj)$, times W_8^1 or W_8^3 can be written by

$$(a+bj) \times \left(\frac{\sqrt{2}}{2}(1-j)\right) = \frac{\sqrt{2}}{2}((a+b) + j(-a+b)) \quad (10)$$

$$(a+bj) \times \left(-\frac{\sqrt{2}}{2}(1+j)\right) = -\frac{\sqrt{2}}{2}((a-b) + j(a+b)) \quad (11)$$

where $\frac{\sqrt{2}}{2}$ can be represented as 1.0101010 using the Booth binary recoded form (BBRF). Thus, the CM unit can be implemented by using simple adders and shifters only. Figure 3 shows the pipelined CM architecture, which uses three subtractions/additions and therefore improves on the computation speed significantly.

Based on the above-mentioned CORDIC arithmetic unit and CM unit, the computational circuit and hardware architecture of the CORDIC-based split-radix 2/8 FFT butterfly computation are realized. As one can see, the pipelined CORDIC arithmetic unit aims at increasing the throughput of complex multiplications..

3.2 ROM-Free Twiddle Factor Generator

In the conventional FFT processor, a large ROM space is needed to store all the twiddle factors. To reduce the chip area, a twiddle factor generator is thus proposed. Figure 4 shows the ROM-free twiddle factor generator using simple adders and shifters for 128-point FFT. In which, the 16-bit accumulator is to generate the value $2n\pi$ for each index n ; $n = 2^{\log_2^N - 3} - 1$, the 16-bit shifter is to divide $2n\pi$ by N , and the 16-bit shifter/adder is to produce the twiddle factors: θ_N^{1n} ,

θ_N^{3n} , θ_N^{5n} and θ_N^{7n} . By using the twiddle factor generator, the chip area and power consumption can be reduced significantly at the cost of an additional logic circuit. Table 1 shows the gate counts of the full-ROM storing all the twiddle factors, the CORDIC twiddle factor generator [1] and the ROM-free twiddle factor generator.

4 Implementation of FFT Processors

The 128/256/512/1024/2048/4096/8192- point FFT processors. In which, the radix-2 and split-radix 2/4 butterfly processors [1] using the pipelined CORDIC arithmetic units and twiddle factor generators are implemented; and moreover, two memory banks (4096/2048/1024/512/256/0 \times 32-bit and 8192/4096/2048/1024/512/256/128 \times 32-bit) are allocated for increased efficiency by using the in-place computation algorithm [1]. Hardware architecture is shown in Figure 5.

The hardware code written in Verilog[®] is running on a workstation with the ModelSim[®] simulation tool and Synopsys[®] synthesis tool (design compiler). The chips are synthesized by the TSMC 0.18 μm 1p6m CMOS cell libraries [18]. The physical circuit is synthesized by the Astro[®] tool. The circuits are evaluated by DRC, LVS and PVS [19].

The layout view of the 8192-point FFT processor is shown in Figure 6. The core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point and 8192-point FFT processors are shown in Table 2. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

5 Performance Analysis of The Proposed FFT Architecture

FFT processors used to compute 128/256/512/1024/ 2048/4096/8192-point FFT are composed mainly of the 128-point CORDIC-based split-radix 2/8 FFT core; the computation complexity using a single 128-point FFT core is $O(N/6)$ for N -point FFT. The log-log plot of the CORDIC computations versus the number of FFT points is shown in Figure 7. As one can see, the proposed FFT architecture is able to improve the power consumption and computation speed significantly.

6 Conclusion

This report presents low-power and high-speed FFT processors based on CORDIC and split-radix techniques for OFDM systems. The architectures are mainly based on a reusable IP 128-point CORDIC-based split-radix FFT core. The pipelined CORDIC arithmetic unit is used to compute the complex multiplications involved in FFT, and moreover the required twiddle factors are obtained by using the proposed ROM-free twiddle factor generator rather than storing them in a large ROM space.

The CORDIC-based 128/256/512/1024/2048/4096/8192- point FFT processors have been implemented by 0.18 μm CMOS, which take 395 μs , 176.8 μs , 77.9 μs , 33.6 μs , 14 μs , 5.5 μs and 1.88 μs to compute 8192-point, 4096-point, 2048-point, 1024-point, 512-point, 256-point and 128-point FFT, respectively.

The CORDIC-based FFT processors are designed by using the portable and reusable Verilog[®]. The 128-point FFT core is a reusable intellectual property (IP), which can be implemented in various processes and combined with an efficient use of hardware resources for the trade-offs of performance, area, and power consumption.

References:

- [1] **T. Y. Sung**, "Memory-efficient and high-speed split-radix FFT/IFFT processor based on pipelined CORDIC rotations," *IEE Proc.-Vis. Image Signal Proc.*, Vol. 153, No. 4, Aug. 2006, pp.405-410.

- [2] J. C. Kuo, C. H. Wen, A. Y. Wu, "Implementation of a programmable 64/spl sim/2048-point FFT/IFFT processor for OFDM-based communication systems," *Proceedings of the 2003 International Symposium on Circuits and Systems*, Volume 2, 25-28 May 2003 pp.II-121 - II-124.
- [3] L. Xiaojin, Z. Lai, C. J. Cui, "A low power and small area FFT processor for OFDM demodulator," *IEEE Transactions on Consumer Electronics*, Volume 53, Issue 2, May 2007, pp. 274 – 277.
- [4] J. Lee, H. Lee, S. I. Cho, S. S. Choi, "A high-speed, low-complexity radix-216 FFT processor for MB-OFDM UWB systems," *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, May 2006.
- [5] A. Cortes, I. Velez, J. F. Sevillano, A. Irizar, "An approach to simplify the design of IFFT/FFT cores for OFDM systems," *IEEE Transactions on Consumer Electronics*, Volume 52, Issue 1, Feb. 2006, pp.26 – 32.
- [6] Y. H. Lee, T. H. Yu, K. K. Huang, A. Y. Wu, "Rapid IP design of variable-length cached-FFT processor for OFDM-based communication systems," *IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 2006 pp.62-65.
- [7] C. L. Wey, W. C. Tang, S. Y. Lin, "Efficient memory-based FFT architectures for digital video broadcasting (DVB-T/H)," *2007 International Symposium on VLSI Design, Automation and Test*, 25-27 April 2007, pp.1-4.
- [8] Y. W. Lin, H. Y. Liu, C. Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," *IEEE Journal of Solid-State Circuits*, Volume 40, Issue 8, Aug. 2005, pp.1726-1735.
- [9] C. D. Thompson, "Fourier transform in VLSI," *IEEE Transactions on Computers*, Vol.32, No. 11, 1983, pp.1047-1057.
- [10] E. H. Wold, A. M. Despain, "Pipelined and parallel-pipelined FFT processor for VLSI implementation," *IEEE Transactions on Computers*, Vol.33, No. 5, 1984, pp.414-426.
- [11] T. Widhe, "Efficient implementation of FFT processing elements," *Linkoping Studies in Science and Technology*, Thesis No. 619, Linkoping University, Sweden, 1997.
- [12] P. Duhamel, H. Hollmann, "Implementation of "split-radix" FFT algorithms for complex, real, and real symmetric data." *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Volume 10, April 1985, pp.784 – 787.
- [13] A .A. Petrovsky, S. L. Shkredov, "Automatic generation of split-radix 2-4 parallel-pipeline FFT processors: hardware reconfiguration and core optimizations," *2006 International Symposium on Parallel Computing in Electrical Engineering*, pp.181-186.
- [14] S. Bouguezal, M. O. Ahmad, M. N. S. Swamy, "A new radix-2/8 FFT algorithm for length-q/spl times/2/sup m/ DFTs," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume 51, Issue 9, 2004, pp.1723- 1732.
- [15] W. C. Yeh, C. W. Jen, "High-speed and low-power split-radix FFT." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 51, Issue 3, March 2003, pp.864 – 874.
- [16] M. D. Ercegovac, T. Lang, "CORDIC algorithm and implementations." *Digital Arithmetic*, Morgan Kaufmann Publishers, 2004, Chapter 11.
- [17] **T. Y. Sung**, H. C. Hsin, "Fixed-point error analysis of CORDIC arithmetic for special-purpose signal processors," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E90-A, No.9, Sep. 2007, pp.2006-2013.

- [18] “TSMC 0.18 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.
- [19] Cadence design systems: <http://www.cadence.com/products/pages/default.aspx>.

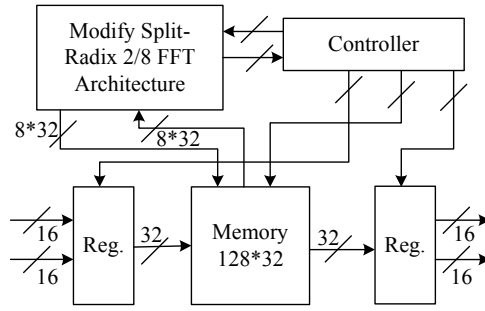


Fig. 1 The proposed 128-point CORDIC- based split-radix FFT processor

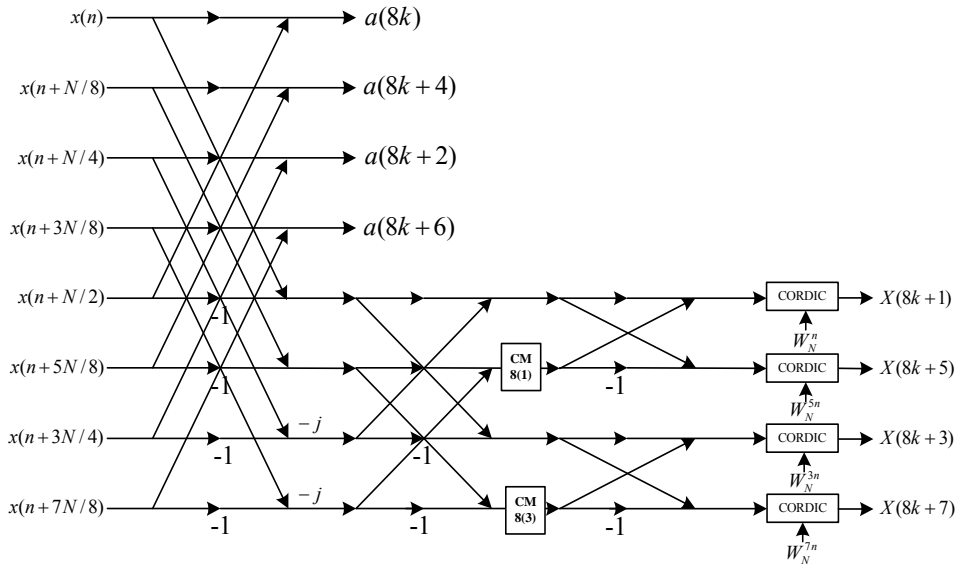


Fig. 2 Data flow of the butterfly computation of the modified split-radix 2/8 FFT

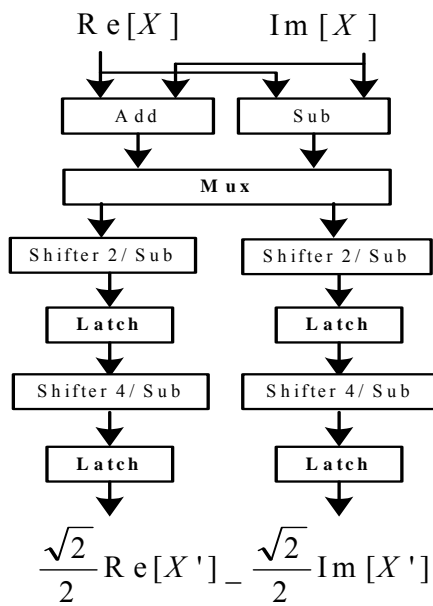


Fig. 3 Constant multiplier (CM) architecture for the modified split-radix 2/8 FFT

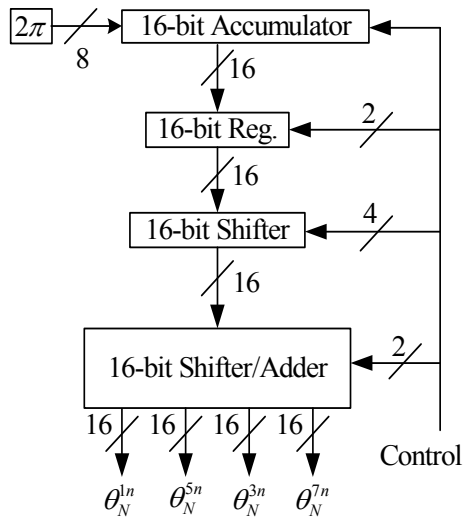


Fig. 4 Proposed ROM-free twiddle factor generator for 128-point FFT

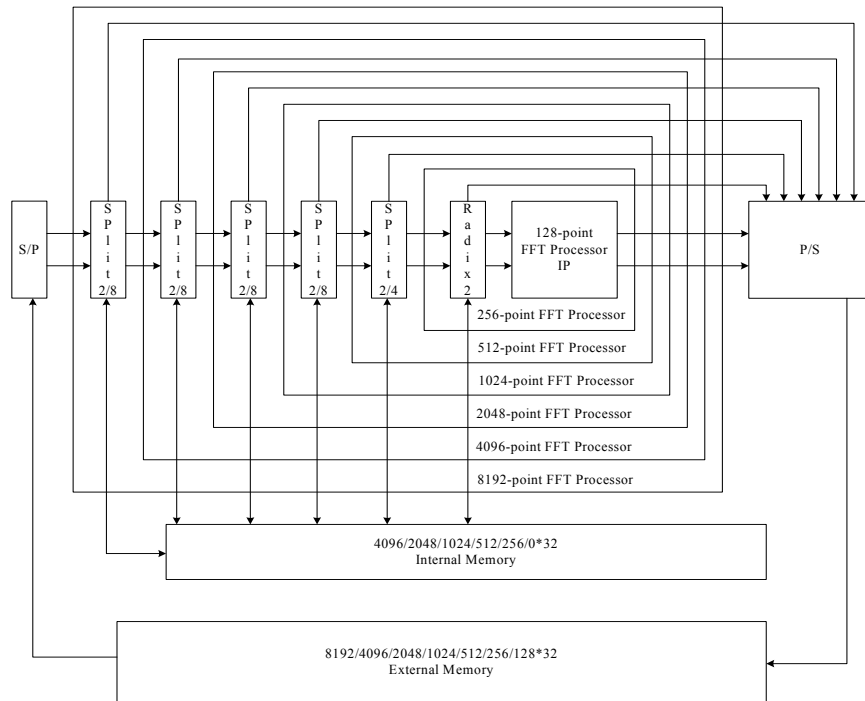


Fig. 5 Hardware architecture of the 128/256/512/1024/2048 /4096 /8192-point FFT processor

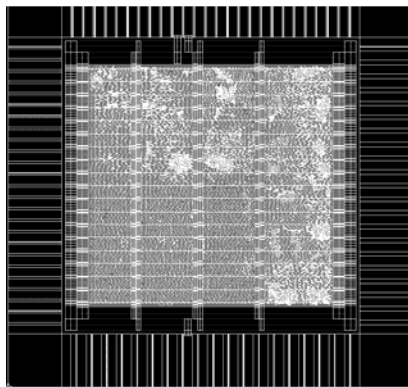


Fig. 6 Layout view of the 8192-point FFT

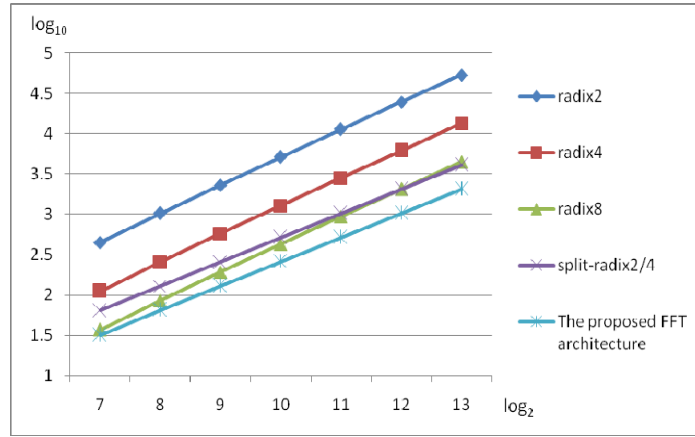


Fig. 7 Log-log plot of the CORDIC computations versus the number of FFT points

Table 1 Hardware requirements of the full-ROM, the CORDIC twiddle factor generator [1], and the ROM-free twiddle factor generator

Full-Twiddle Factor ROM					1 bit ~ 1 gate
8192-Point ROM					
4K × 16 bit					
CORDIC Twiddle Factor Generator			(T. Y. Sung, 2006) [1]		
16-bit CORDIC ~ 18K bit	11-bit Adder ~ 150 gates	11-bit Shifter ~ 50 gates	16-bit Shifter ~ 90 gates	16-bit Adder ~ 200 gates	
ROM-free Twiddle Factor Generator (Sung, Hsin and Cheng, 2008)					
16-bit Accumulator ~ 200 gates	16-bit Register ~ 32 gates	16-bit Shifter ~ 90 gates	16-bit Shifter/Adder ~ 90 × 2 + 200 × 2 gates		

Table 2 Core areas, power consumptions, clock rates of 128-, 256-, 512-, 1024-, 2048-, 4096- and 8192-point FFT processors

FFT Size	Core Area	Power Consumption	Clock Rate
128-point	2.28mm ²	80mW	200MHz
256-point	2.37mm ²	84mW	200MHz
512-point	2.49mm ²	88mW	200MHz
1024-point	2.62mm ²	94mW	200MHz
2048-point	2.81mm ²	99mW	200MHz
4096-point	3.10mm ²	106mW	200MHz
8192-point	3.62mm ²	117mW	200MHz

1 Introduction

With the rapid growth of modern communication applications and computer technologies, image compression continues to be in great demand. Three categories of image compression techniques have been developed: differential pulse code modulation, transform coding and subband coding. In many cases, transform coding is preferable. The simplest transform coding approach is based on Walsh-Hadamard transform, in which the kernel matrix involves simple additions and subtractions only [1]. Karhunen-Loeve transform (K-LT) is the optimal; however the computation involved is too complicated to be implemented. Discrete cosine transform (DCT), which approximates well K-LT [2], is efficient and therefore adopted by the Joint Photographic Expert Group (JPEG) standard. Many DCT algorithms have been developed [3]-[11]; the VLSI implementations of DCT for real-time applications can be found in [12]-[25].

Though fast Fourier transform (FFT) can be utilized to implement DCT, it requires complex-valued computations. In addition, the order of N -point DCT through the use of FFT is $O(\log 2N + 1)$. CORDIC (COordinate Rotation DIgital Computer) is a well known algorithm, which evaluates various elementary functions including sine and cosine functions by using simple adders and shifters. CORDIC is suitable for the design of high performance chips using VLSI technology. In this paper, the CORDIC approach to the development of fast, memory-efficient DCT and IDCT is proposed to simplify hardware implementations and reduce power consumption.

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 [26]-[27]

$$x_{i+1} = x_i - \sigma_i 2^{-i} y_i \quad (1)$$

$$y_{i+1} = y_i + \sigma_i 2^{-i} x_i \quad (2)$$

$$z_{i+1} = z_i - \sigma_i \alpha_i \quad (3)$$

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

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

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

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

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.

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

Thus, the convergence range of angles is expanded to cover a whole plane of $[-180^\circ, 180^\circ]$, in other words, the input angle is unconstrained [28]-[29].

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

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

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 8×8 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 8×8 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

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

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 $5\pi/16$. The details of blocks CORDIC(1), CORDIC(2) and CORDIC(3) are shown in Figure 2.

3.2 Fast 1-D IDCT Algorithm

Similarly, the fast 8-point IDCT can be obtained by further decomposing Matrix \mathbf{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 3 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 $6\pi/16$. The details of blocks R0 and R1 are shown in Figure 4.

3.3 CORDIC-Based 2-D DCT and IDCT Architectures

The demanding operation of both DCT and IDCT, i.e. multiplication can be simplified significantly by using CODIC. Specifically, multipliers required for the implementation of both DCT and IDCT can be replaced by simple shifters and adders through the use of CORDIC processor. In which, each CORDIC operates in the circular coordinate system with some fixed angle that can be pre-decomposed into a sequence of micro-rotations, $\{\sigma_i\}$, and can be stored in ROM. Based on constant-rotation CORDIC [30]-[31], multiplier involved in DCT and IDCT can be implemented by using two shifters and two adders. Figure 5 depicts a constant-rotation CORDIC in the circular coordinate system. It is noted that each CORDIC processor used as the arithmetic unit (AU) in the proposed fast DCT/IDCT architectures can save hardware by one third to achieve low-power consumption.

4 The Proposed 2-D DCT and IDCT Processors

Based on equations (11) and (15), an efficient pipelined architecture has been developed for 2D DCT/IDCT. Figure 6 shows the proposed architecture for 8×8 DCT and IDCT processors, where one 64-word SRAM bank, two 8-point DCT/IDCT processors and a control unit are used. The 8-point 1-D DCT/IDCT input-processor denoted by P1 writes the intermediate result into the row and column of SRAM bank alternately. P2 denotes the 8-point 1-D DCT/IDCT output-processor, which first reads data from the column and raw of SRAM bank alternately and then outputs the final result. Figure 7 shows the finite state machine (FSM) of the control unit, which manages data flow and timing for 2-D DCT/IDCT operations.

4.1 Implementation of 1-D DCT/IDCT Processor

Constant-rotation CORDIC processors are used to implement 8-point DCT and IDCT processors, which are shown in Figure 8 and Figure 9, respectively. Note that the transform matrices of DCT and IDCT are column symmetry and row symmetry, respectively, the shuffle structures are simplified and no multipliers are utilized.

4.2 Implementation of 2-D DCT/IDCT Processors

One of the crucial issues using the conventional DCT/IDCT architectures with single processor is long latency and low throughput [6]-[8]. Moreover, the conventional DCT/IDCT architectures with single memory bank can not be pipelined [21]-[25], [29]. In [18]-[20], Sung proposed a pipelined architecture with dual memory banks to double the throughput of DCT/IDCT at the cost of increasing memory space. To save memory space while increasing throughput, we propose

pipelined 2-D DCT/IDCT architecture with single memory bank, which is shown in Figure 6; The latency of 1-D DCT/IDCT processor is 8 clocks, hardware complexity is of order $O(N \cdot \log_2 N)$, and throughput is 8 outputs per cycle. Note that multiplier is replaced by simple adders and shifters based on constant-rotation CORDIC so that many desirable properties, e.g. small area, low-power and high throughput can be obtained. Specifically, the proposed 2-D DCT/IDCT architecture can improve throughput by two times that of the conventional architectures and save memory space by 50%. Table 1 shows comparisons between the proposed architecture and the conventional architectures [6]-[8], [18]-[20] with dual memory banks, and [21]-[25]. Table 2 shows comparisons with other commonly used architectures [9]-[12].

The proposed pipelined architecture for 32-bit fixed-point 2-D DCT and IDCT processors have been written in Verilog[®] and synthesized by TSMC 0.18 μm 1P6M CMOS cell libraries [32]. Core size and power consumption can be obtained from the reports of Synopsys[®] design analyzer and PrimPower[®] [33], respectively. The core sizes of the proposed 2-D DCT and IDCT processors are $2372 \times 2372 \mu\text{m}^2$ and $2396 \times 2396 \mu\text{m}^2$, respectively; their respective 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. The layout views of the implemented 2-D DCT and IDCT processors are shown in Figure 10 and Figure 11, respectively, which are much suited to the JPEG and MPEG applications.

Due to financial problem, the platform for architecture development and verification has been implemented; the architecture is implemented on Xilinx XC2V6000 FPGA emulation board [34] with an 8051 microcontroller and interface circuits (USB 2.0) [35] as shown in Figure 12. This 8051 microcontroller uses USB 2.0 bus to read the original image from PC through DMA channel and write the processed image back to PC. The Xilinx XC2V6000 FPGA chip implements DCT and IDCT. Figure 13 shows the architecture development and verification board. The original 512×512 Lena image is shown in Figure 14; the reconstructed image is shown in Figure 15. 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

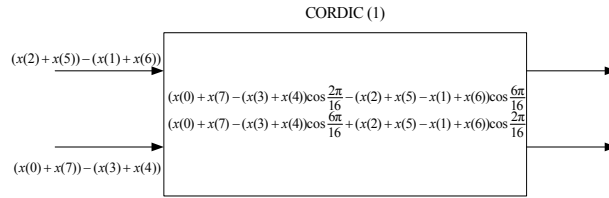
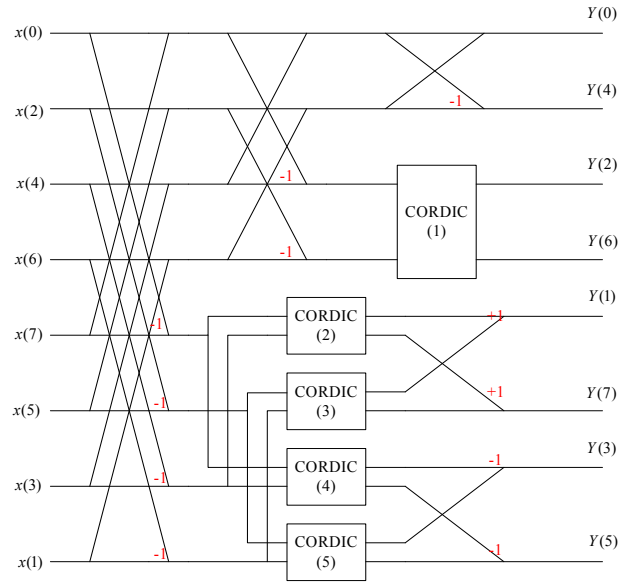
performances, area and power consumption trade-offs. The proposed 2-D DCT and IDCT processors are much suited to the applications of JPEG, MPEG-4 and H.264 standards.

References:

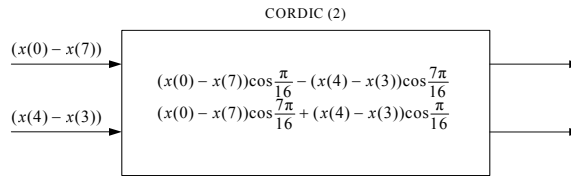
- [1] D. F. Elliott, K. R. Kao, *Fast Transforms Algorithms, Analysis, Applications*, Chapter 8, Walsh-Hadamard Transform, Prentice-Hall, 1982, pp.301-303.
- [2] R. J. Clarke, Relation between the Karhunen Loeve and Cosine Transform," *IEEE Proceedings*, Part F, vol. 128, no. 6, Nov. 1981, pp.359-360.
- [3] M. J. Narasimha, A. M. Peterson, On the Computation of the Discrete Cosine Transform, *IEEE Transactions on Communications*, vol. 26, no. 6, June 1978, pp. 934-936.
- [4] R. M. Haralick "A Storage Way to Implement the Discrete Cosine Transform," *IEEE Transactions on Computers*, July 1976, pp.764-765.
- [5] W. H. Chen, C. H. Smith, S. C. Fralick, "Fast Computational Algorithm for the Discrete Cosine Transform," *IEEE Transactions on Communications*, vol. 25, no. 9, Sept. 1977, pp.1004-1009.
- [6] **T. Y. Sung**, VLSI Parallel and Distributed Computation Algorithms for DCT Processors, *Proceedings IEEE International Phoenix Conference on Computer and Communications, Scottsdale, Arizona, USA*, 1990, pp.121-125.
- [7] **T. Y. Sung**, 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*, March 1990, pp.36-39.
- [8] **T. Y. Sung**, Novel Parallel VLSI Architectures for Discrete Cosine Transforms, *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, USA*, April 1990, pp.998-1001.
- [9] 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 Transactions on Circuits Systems for Video Technology*, vol. 7, no. 1, June 1997, pp.459-467.
- [10] Y. T. Chang, C. L. Wang, 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, April 1995, pp.150-157.
- [11] S. F. Hsiao, W. R. Shiue, 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, Nov. 2001, pp.1149-1159.
- [12] 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.
- [13] 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. Circuits and Systems, Part-I: Regular Papers*, vol. 52, no. 8, Aug. 2005, pp.1568-1579.
- [14] V. Srinivasan, K. J. R. Liu, 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, Feb. 1996, pp.87-96.
- [15] T. Kuroda, 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, Nov.

- 1996, pp.1770-1778.
- [16] R. Rambaldi, A. Uguzzoni, R. Guerrieri, A 35 μ W 1.1 V Gate Array 8 \times 8 IDCT Processor for Video-Telephony, *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998, pp.2993-2996.
- [17] T. H. Chen, A Cost-Effective 8 \times 8 2-D IDCT Core Processor with Folded Architecture, *IEEE Transactions on Consumer Electronics*, vol. 45, no.2, May 1999, pp.333-339.
- [18] **T. Y. Sung**, Y. H. Sung, A Novel Implementation of Cost-Effective Parallel-Pipelined 8 \times 8 DCT Processor, *The Fourth IEEE Asia-Pacific Conference on Advanced System Integrated Circuits (AP-ASIC) 2004*, Fukuoka, Japan, August 3-5, 2004, pp.200-203.
- [19] **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 (Algo-2006)*, Changhua, Taiwan, April 28~29, 2006, pp.369-372.
- [20] **T. Y. Sung**, M. J. Sun, H. C. Hsin, C. W. Yu, Low-Power and High-Speed Architectures for 2-D DCT and IDCT Based on CORDIC Rotation, *19th Computer Vision, Graphics, and Image Processing Conference*, Taiwan Aug. 13-15, 2006, pp.1024-1029.
- [21] Y. H. Hu, Z. Wu, An Efficient CORDIC Array Structure for the Implementation of Discrete Cosine Transform, *IEEE Transactions on Signal Processing*, vol. 43, no. 1, Jan. 1995, pp.331-336.
- [22] H. Jeong, J. Kim, W. K. Cho, Low-Power Multiplierless DCT Architecture Using Image Data Correlation, *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, Feb. 2004, pp.262-267.
- [23] D. Gong, Y. He, Z. Gao, 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, April 2004, pp. 405-415.
- [24] V. Dimitrov, K. Wahid, G. Jullien, Multiplication-Free 8 \times 8 2D DCT Architecture Using Algebraic Integer Encoding, *Electronics Letters*, vol. 40, no. 20, Sept. 2004, pp.1310-1311.
- [25] M. Alam, W. Badawy, G. Jullien, 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, May 2005, pp.726-730.
- [26] J. E. Volder, The CORDIC Trigonometric Computing Technique, *IRE Transactions on Electronic Computers*, vol. EC-8, 1959, pp.330-334.
- [27] J. S. Walther, A Unified Algorithm for Elementary Functions, *Spring Joint Computer Conference Proceedings*, vol. 38, 1971, pp.379-385.
- [28] X. Hu, R. G. Harber, S. C. Bass, Expanding the range of the Convergence of the CORDIC Algorithm, *IEEE Transactions on Computers*, vol. 40, no. 1, 1991, pp.13-21.
- [29] **T. Y. Sung**, Y. H. Sung, The Quantization Effects of CORDIC Arithmetic for Digital Signal Processing Applications, *The 21st Workshop on Combinatorial Mathematics and Computation Theory*, Taiwan, May 21~22, 2004, pp.16-25.
- [30] **T. Y. Sung**, Y. H. Hu, T. M. Parng, Design and Implementation of a VLSI CORDIC Processor, *Proc. 1986 IEEE Int. Symp. Circuits Syst.*, vol. 3, 1986, pp.934-935.
- [31] **T. Y. Sung**, Y. H. Hu, H. J. Yu, "Doubly Pipelined CORDIC Array for Digital Signal Processing Algorithms," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '86)*, vol. 11, Apr. 1986, pp. 1169-1172.

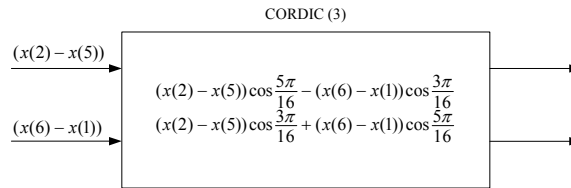
- [32] *TSMC 0.18 μm CMOS Design Libraries and Technical Data*, v.3.2, Taiwan Semiconductor Manufacturing Company, Hsinchu, Taiwan, and National Chip Implementation Center (CIC), National Science Council, Hsinchu, Taiwan, R.O.C., 2006.
- [33] Synopsys products, <http://www.synopsys.com/products>.
- [34] Xilinx FPGA products, <http://www.xilinx.com/products>.
- [35] **T. Y. Sung**, C. W. Yu, Y. S. Shieh, A High-Efficient and Cost-Effective LCD Signal Processor, *7th International Conference on Computer Vision, Pattern Recognition and Image Processing*, Taiwan, 2006, pp.939-942.



(a)



(b)



(c)

Fig. 2. The details of blocks (a) CORDIC(1), (b) CORDIC(2) and (c) CORDIC (3)

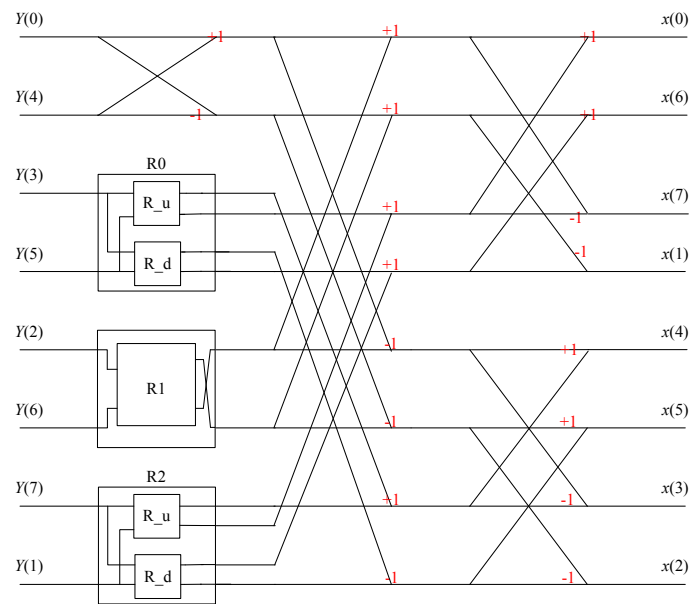


Fig.3. Data flow of 8-point IDCT

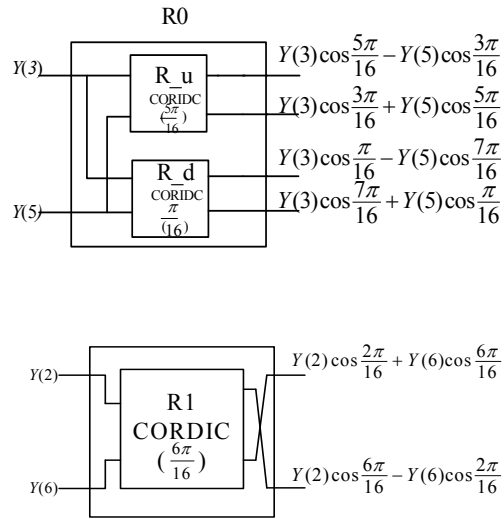


Fig. 4. The details of blocks (a) R0 and (b) R1

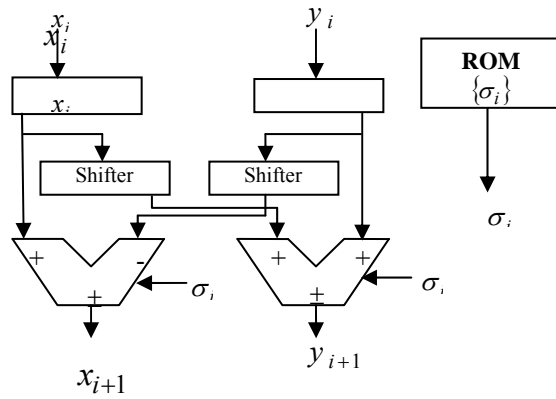


Fig. 5. Constant-rotation CORDIC in the circular coordinate system (Reduced CORDIC)

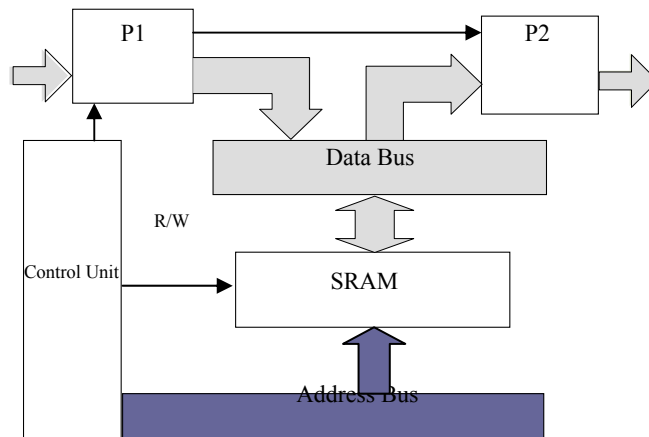


Fig.6. The proposed architecture for 2-D DCT/IDCT processor (P1 and P2: 1-D DCT/IDCT processor)

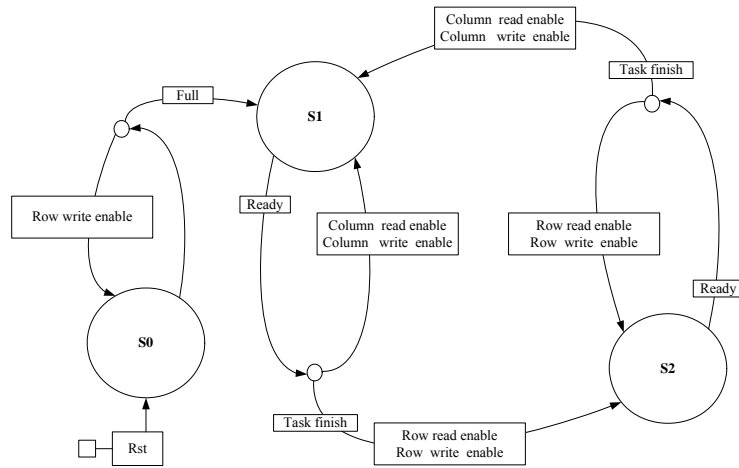


Fig.7. The finite state machine (FSM) of the control unit

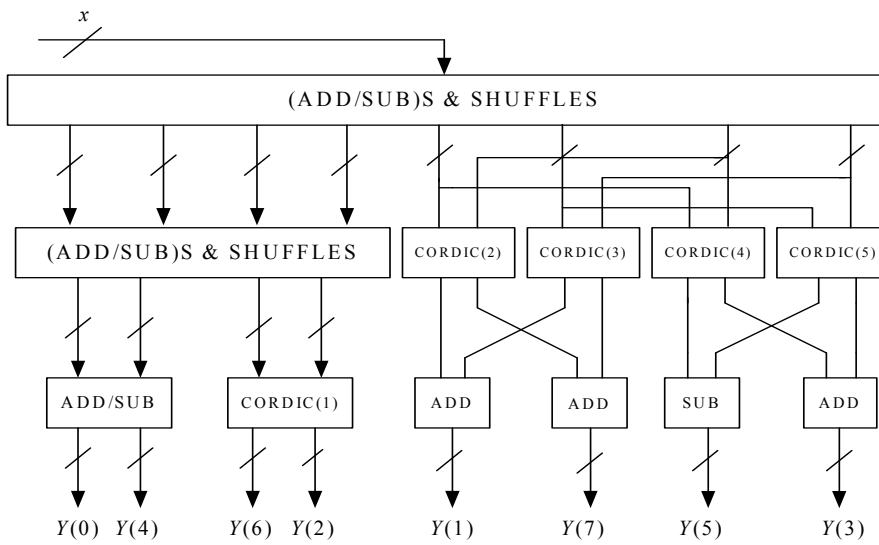


Fig.8. The proposed 1-D 8-point DCT processor

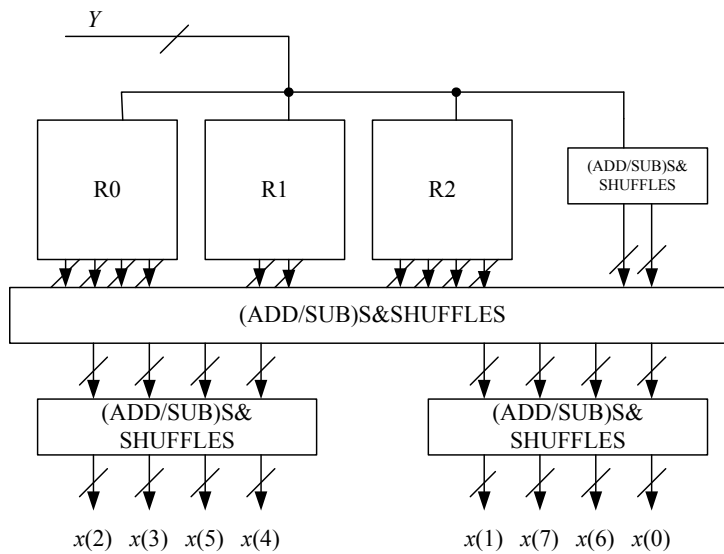


Fig.9. The proposed 1-D 8-point IDCT processor

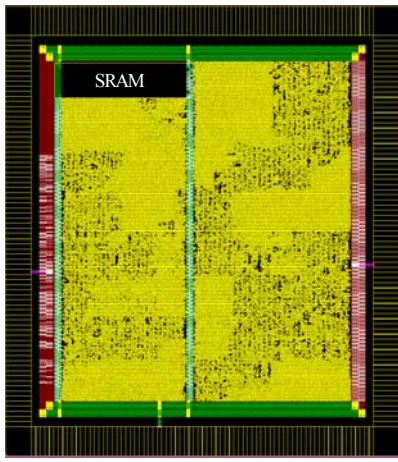


Fig.10. The layout view of the implemented 2-D DCT processor

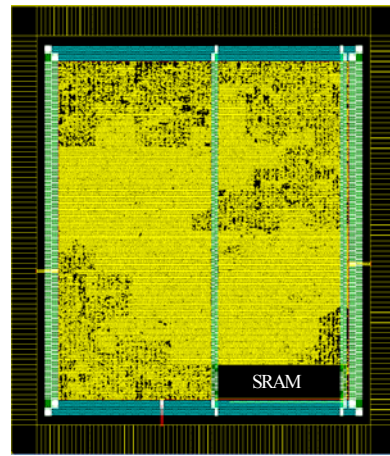


Fig.11. The layout view of the implemented 2-D IDCT processor

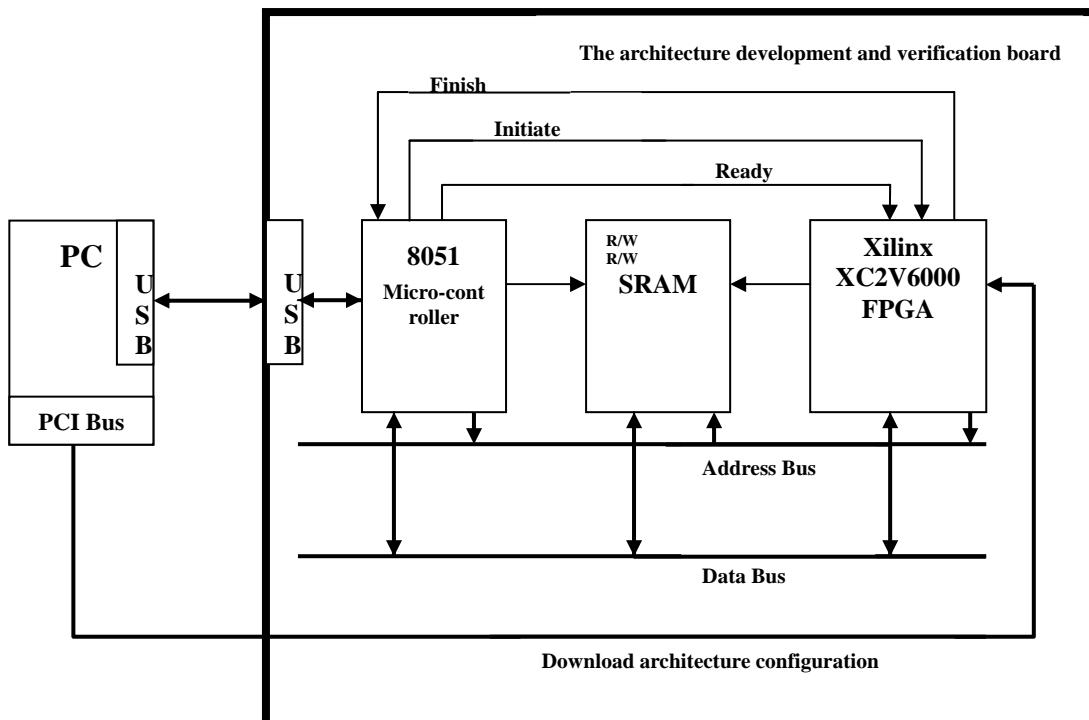


Fig.12. Block diagram of the architecture development and verification platform for DCT and IDCT

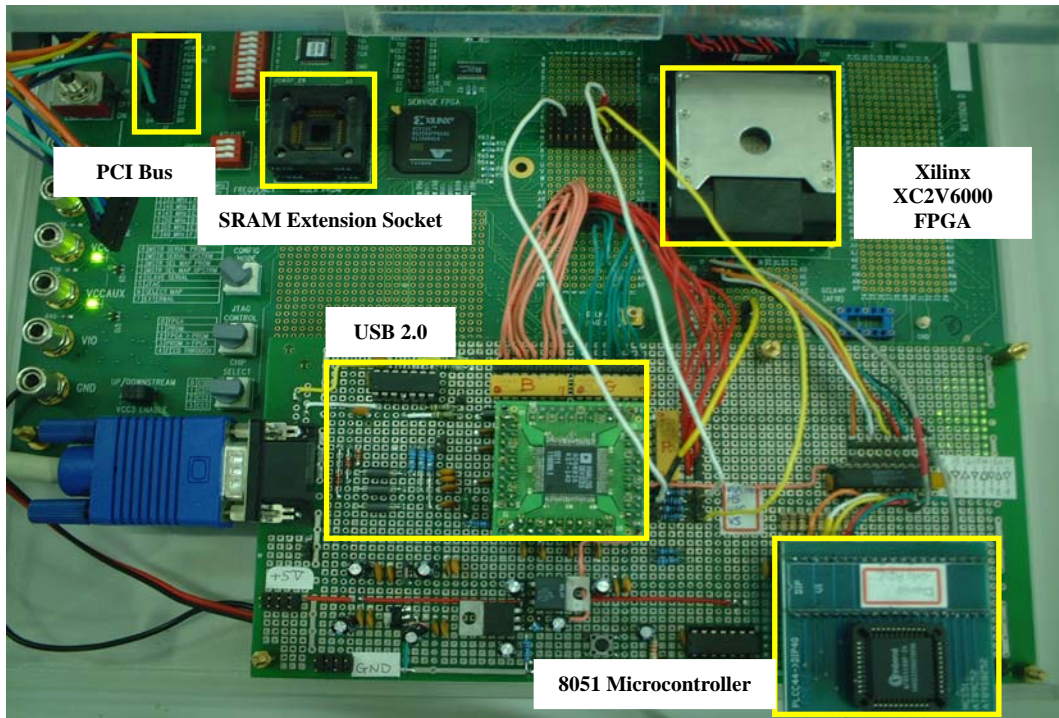


Fig.13. The architecture development and verification board

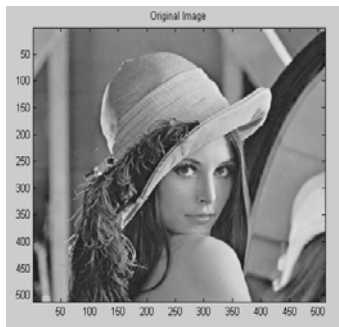


Fig.14 Original image

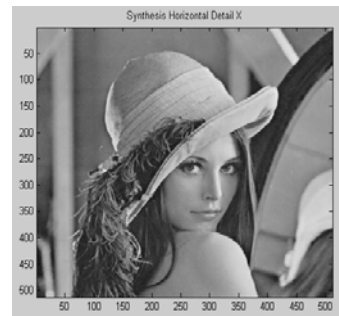


Fig.15 Reconstructed image

Table 1 Comparison between the proposed architectures and the conventional architectures

8×8 2-D DCT/IDCT	The conventional architectures		The proposed pipelined architectures	
	The single-processor architectures [6]-[8]	The parallel architectures with single memory-bank [21]-[25]	The pipelined architectures with dual memory-bank [18]-[20]	This work
Processors	1	8	5 (Complete CORDIC)	5 (Reduced CORDIC)
Real-multipliers	2	16	0	0
Real-Adders	3	18	18	18
RAM	64	64	134	64
ROM	6	6	6	6
Hardware complexity (AUs)	$O(1)$	$O(N - \log_2 N + 1)$	$O(N - \log_2 N)$	$O(N - \log_2 N)$
Computation complexity	$O(N^2)$	$O(2N)$	$O(N)$	$O(N)$
Latency	64	16	8	8
Throughput (outputs/cycle)	1	4	8	8
Hardware utilization (100%)	yes	no	yes	yes
Pipelinenability	no	no	yes	yes
Power consumption	poor	poor	good	better

Table 2 Comparison of the proposed architecture to other commonly used architectures

8×8 2-D DCT/IDCT	Lee [9] DCT/IDCT	Chang [10] DCT/IDCT	Hsiao [11] DCT	Hsiao [12] DCT/IDCT	Hsiao [13] DCT/IDCT	Sung [18]-[20] DCT/IDCT	This Work DCT/IDCT
Real-multipliers	28	64	-	-	-	-	-
CORDIC processors	-	-	-	-	3	5(Complete)	5 (Reduced)
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)	$O(M \log N)$	$O(N^2)$	$O(\log N)$	$O(\log N)$	$O(\log N)$	$O(N - \log N)$	$O(N - \log N)$
Throughput (outputs/cycle)	16	8	2	2	2	8	8
Hardware utilization (100%)	no	no	no	no	no	yes	yes
Pipelinenability	no	no	no	no	yes	yes	yes
Parallelism	yes	yes	yes	yes	yes	yes	yes

計畫成果自評：

本研究之第一部分預期能完成一個充分利用座標旋轉原理演算法的特性以簡化FFT的架構，增加晶片之性能，並且節省參數儲存之記憶體。研究報告中已經報告完成一個改良式重組式超大型積體電路架構FFT，運算因子記憶體也完全省略，所增加之產生器電路非常精簡，晶片面積充分得以減少，晶片耗電也得以節省。運算速度在架構改進下得以增快，已充分達到預期目標。從報告之成果比較表，如下所示中：

Full-Twiddle Factor ROM		1bit~1gate		
8192-Point ROM				
4K×16 bit				
CORDIC Twiddle Factor Generator		(T. Y. Sung, 2006) [1]		
16-bit CORDIC	11-bit Adder	11-bit Shifter	16-bit Shifter	16-bit Adder
~18K bit	~150 gates	~50 gates	~90 gates	~200 gates
ROM-free Twiddle Factor Generator (Sung and Hsin, 2009)				
16-bit Accumulator	16-bit Register	16-bit Shifter	16-bit Shifter/Adder	
~200gates	~32 gates	~90 gates	~90×2 + 200×2 gates	

可充分得知本研究已獲得相當優良的成果。

本研究之第二部分也能達成預期之目標完成一個以座標旋轉原理演算法為基礎之高效率超大型積體電路架構之順向及反向二維正弦轉換處理器。在記憶體及速度上均有特別的改進，在功率削好上也得到很低的功耗。報告之成果比較表中得以充分的說明，比較表如下圖所示：

Table 1 Comparison between the proposed architectures and the conventional architectures

8×8 2-D DCT/IDCT	The conventional architectures		The proposed pipelined architectures	
	The single-processor architectures [6]-[8]	The parallel architectures with single memory-bank [21]-[25]	The pipelined architectures with dual memory-bank [18]-[20]	This work
Processors	1	8	5 (Complete CORDIC)	5 (Reduced CORDIC)
Real-multipliers	2	16	0	0
Real-Adders	3	18	18	18
RAM	64	64	134	64
ROM	6	6	6	6
Hardware complexity (AUs)	$O(1)$	$O(N - \log_2 N + 1)$	$O(N - \log_2 N)$	$O(N - \log_2 N)$
Computation complexity	$O(N^2)$	$O(2N)$	$O(N)$	$O(N)$
Latency	64	16	8	8
Throughput (outputs/cycle)	1	4	8	8
Hardware utilization (100%)	yes	no	yes	yes
Pipelability	no	no	yes	yes
Power consumption	poor	poor	good	better

Table 2 Comparison of the proposed architecture to other commonly used architectures

8×8 2-D DCT/IDCT	Lee [9] DCT/IDCT	Chang [10] DCT/IDCT	Hsiao [11] DCT	Hsiao [12] DCT/IDCT	Hsiao [13] DCT/IDCT	Sung [18]-[20] DCT/IDCT	This Work DCT/IDCT
Real-multipliers	28	64	-	-	-	-	-
CORDIC processors	-	-	-	-	3	5(Complete)	5 (Reduced)
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
Pipelability	no	no	no	no	yes	yes	yes
Parallelism	yes	yes	yes	yes	yes	yes	yes

本研究之第一部分已經投稿Journal of Digital Signal Processing (SCI)國際著名期刊，已進行修改(Revise)，可望在今年登出。同時計畫申請台灣及美國發明專利，以取得矽智產權。本研究在實用面上，可應用於無線寬頻網絡晶片，為必要的核心功能單元。

本研究之第二部分已透過中華大學申請台灣專利，此一研究之成果可應用於H.264相關之晶片電路。目前並已計畫申請美國專利，且相關研究將投稿國際期刊。

本年度專題研究計畫中，獲得相關心得很多，以座標旋轉原理演算法應用於無線寬頻網絡上之機會日益增加，尚有許多研究可以進行，因此於98年度國科會專題研究計畫中即提出直接數位頻率合成器(DDFS-Direct Digital Frequency Synthesizer)之研究，已獲通過(計畫案號: NSC98-2221-E-216-037)，計畫為期一年。未來相關無線寬頻網絡的電路研究將持續進行，將會有一系列之成果產生。

總結本年度研究計畫，已有相當的成果，除達到預期研究目標外，實際成果超越當初的期望，在研究期間即投稿國際期刊，並獲佳評，預料論文經過少部分修改後，即將於期刊登出。同時，兩個研究成果除第二部分研究已申請專利外，第一部分研究將循先前之相關研究申請美國專利的腳步，將繼續透過中華大學申請台灣及美國專利。

可供推廣之研發成果資料表(一)

■ 可申請專利

■ 可技術移轉

日期：98年7月20日

<p>國科會補助計畫</p>	<p>計畫名稱：快速座標旋轉原理演算法應用於特殊數位信號處理之晶片設計與製作</p> <p>計畫主持人：宋志雲</p> <p>計畫編號：NSC 97-2221-E-216-044</p> <p>學門領域：積體電路及系統設計</p>
<p>技術/創作名稱</p>	<p>可重組式快速傅利葉轉換之超大型積體電路架構</p>
<p>發明人/創作人</p>	<p>宋志雲、辛錫進</p>
<p>技術說明</p>	<p>中文：</p> <p>本研究第一部分報告說明一個座標旋轉原理(CORDIC)為基礎之分離式基底快速傅利葉轉換(split-radix fast Fourier transform)核心用於正交頻分複用技術(OFDM)，例如超寬頻無線網路(UWB)，非對稱數位用戶線路(ADSL)，數位音訊廣播(DAB)，數位視訊廣播-陸上系統(DVB-T)，高速數位用戶線路(VHDSL)以及全球互通微波存取(WiMAX)。高速 128/256/512/1024/ 2048/4096/8192 點快速傅利葉轉換處理器於本研究中以台積電製程 0.18 微米(1p6m)完成晶片設計，所有控制電路均集成於單一晶片上。所完成之快速傅利葉轉換處理器在功率消耗上以及晶片面積上，均獲得較佳之效率。</p> <p>關鍵詞-矽智產權(Intellectual Property)，快速傅利葉轉換，分離式基底，座標旋轉原理，正交頻分複用技術。</p> <p>The first part of report presents a Coordinate Rotation Digital Computer (CORDIC)-based split-radix fast Fourier transform (FFT) core for Orthogonal Frequency Division Multiplexing (OFDM) systems, for example, Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL), and Worldwide Interoperability for Microwave Access (WiMAX). High-speed 128/256/512/1024/ 2048/4096/8192-point FFT processors have been implemented by 0.18 μm (1p6m) at 1.8V, in which all the control signals are generated internally. These FFT processors outperform the conventional ones in terms of both power consumption and core area.</p> <p><i>Key-Words:</i> - Intellectual Property (IP), FFT, split-radix, CORDIC, OFDM.</p>

可利用之產業 及 可開發之產品	無線網路設備相關之晶片組。
技術特點	省電電路，快速運算，晶片面積節約。
推廣及運用的價值	有機會廣泛利用在無線網路(WiMAX)相關之晶片設計。將申請台灣及美國專利。

- ※ 1. 每項研發成果請填寫一式二份，一份隨成果報告送繳本會，一份送 貴單位研發成果推廣單位（如技術移轉中心）。
- ※ 2. 本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。
- ※ 3. 本表若不敷使用，請自行影印使用。

可供推廣之研發成果資料表(二)

■ 可申請專利

■ 可技術移轉

日期：98年7月20日

<p>國科會補助計畫</p>	<p>計畫名稱：快速座標旋轉原理演算法應用於特殊數位信號處理之晶片設計與製作</p> <p>計畫主持人：宋志雲</p> <p>計畫編號：NSC 97-2221-E-216-044</p> <p>學門領域：積體電路及系統設計</p>
<p>技術/創作名稱</p>	<p>高效率以座標旋轉演算法為基礎之二維順向及反向正弦轉換處理器</p>
<p>發明人/創作人</p>	<p>宋志雲、辛錫進</p>
<p>技術說明</p>	<p>中文：</p> <p>本研究報告第二部分說明一個超大型積體電路架構之二維順向及反向8×8正弦轉換處理器，此一處理器的架構具備平行及疊流結構，其中包含64位元之靜態隨機記憶體，6個位元的唯讀記憶體置放參數，與傳統演算法之架構相比較，節省記憶體空間。此一處理器中的乘法器悉數被座標旋轉原理處理器取代，節省許多硬體，同時減少功率消耗，增加多效率。</p> <p>關鍵詞：- 順向及反向正弦轉換，平行及疊流結構，低功率，座標旋轉原理。</p> <p>Two-dimensional discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) have been widely used in many image processing systems. In the second part of report, 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.</p> <p>Key-Words: - DCT, IDCT, parallel and pipelined architecture, low CORDIC.</p>
<p>可利用之產業 及 可開發之產品</p>	<p>影像處理之晶片組</p>

技術特點	省電電路，快速運算，晶片面積節約。
推廣及運用的價值	有機會廣泛利用在影像壓縮 H.264 相關之晶片設計。已申請台灣專利及將申請美國專利。

- ※ 1. 每項研發成果請填寫一式二份，一份隨成果報告送繳本會，一份送 貴單位研發成果推廣單位（如技術移轉中心）。
- ※ 2. 本項研發成果若尚未申請專利，請勿揭露可申請專利之主要內容。
- ※ 3. 本表若不敷使用，請自行影印使用。

附錄：

附件一：研究發表之論文

附件二：出席國際學術會議心得報告及發表之論文

附件一：研究發表之論文

High-SFDR and Multiplierless Direct Digital Frequency Synthesizer

Tze-Yun Sung

Department of
Microelectronics Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
bobsung@chu.edu.tw

Hsi-Chin Hsin

Department of Computer
Science and Information
Engineering
National United University
Miaoli 36003, Taiwan
hsin@nuu.edu.tw

Lyu-Ting Ko

Department of Electrical
Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
m09601049@chu.edu.tw

Abstract: - This paper presents a hybrid CORDIC (COordinate Rotation DIgital Computer) algorithm for designs and implementations of the direct digital frequency synthesizer (DDFS). The proposed multiplier-less architecture with small ROM (4×16-bit) and pipelined data path provides a spurious free dynamic range (SFDR) of more than 84.4 dBc. A SoC (system on chip) has been designed by 1P6M CMOS, and then emulated on the Xilinx FPGA. It is shown that the hybrid CORDIC-based architecture is suitable for VLSI implementations of the DDFS in terms of hardware cost, power consumption, and SFDR.

Key-Words: - DDFS, hybrid CORDIC, SoC, FPGA, SFDR.

1 Introduction

The direct digital frequency synthesizer (DDFS) plays a key role in many digital communication systems. Fig. 1 depicts the conventional DDFS, which consists mainly of phase accumulator, sine/cosine generator, digital-to-analog converter, and low-pass filter. The sine/cosine generator as the core of DDFS is usually implemented by using a ROM lookup table; with high spurious free dynamic ranges (SFDR) comes a large ROM lookup table [1]. In order to reduce the size of the lookup table, many techniques were proposed [1]-[4]. The quadrant compression technique can reduce the ROM size by 75% [2]. The Sunderland architecture is to split the ROM into two smaller ones [3], and its improved version known as the Nicholas architecture results in a higher ROM-compression ratio (32:1) [4]. In [5], the polynomial hyperfolding technique with high order polynomial approximation was used to design DDFS. In [6] and [7], the angle rotation algorithm was used to design quadrature direct digital frequency synthesizer/complex mixer (QDDSM). COordinate Rotation DIgital Computer (CORDIC) is a well known arithmetic algorithm, which evaluates various elementary functions including sine and cosine functions by using simple adders and shifters only. Thus, CORDIC is suitable for the design of high-performance chips with VLSI technologies. Recently, the CORDIC algorithm has received a lot of attention to the design of high-performance DDFS [8]-[11], especially for the modern digital

communication systems.

This paper is organized as follows. In section II, the hybrid CORDIC algorithm is proposed. In section III, hardware implementation of DDFS is described. The performance analysis is presented in section IV. Finally, the conclusion is given in section V.

2 The Hybrid CORDIC Algorithm

In this section, the hybrid CORDIC algorithm is proposed, and based on which, a low-power and high-SFDR DDFS can be developed.

2.1 Modified Angle Recoding Method for CORDIC Algorithm

In order to reduce the number of CORDIC iterations, the input angle can be divided into encoded angles by using the modified Booth encoding (MBE) method [12]. Specifically, let ψ denote the input angle represented by

$$\psi = f(0)2^{-p} + f(1)2^{-(p+1)} + \dots + f(w-1)2^{-(w-1)} \quad (1)$$

where $f(i) \in \{0,1\}$, w is the word length of operands, and $\left\lceil \frac{(w-2.585)}{3} \right\rceil = p \leq i \leq w-1$. The MBE decomposition of ψ is as follows.

$$\psi = \sum_{i=p/2}^{(w-1)/2} \beta(i) \quad (2)$$

where the encoded angle:

$$\beta(i) = \rho(i)2^{-i} \text{ with } \rho(i) \in \{-2,-1,0,1,2\}. \text{ As } \sin \beta(i)$$

and $\cos \beta(i)$ can be approximated by

$$\sin \beta(i) \cong \beta(i) = \rho(i)2^{-i} \quad (3)$$

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\cos \beta(i) \cong 1 - \frac{\beta^2(i)}{2} = 1 - \rho^2(i)2^{-(2i+1)} \quad (4)$$

we have

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 - \rho^2(i)2^{-(2i+1)} & \rho(i)2^{-i} \\ -\rho(i)2^{-i} & 1 - \rho^2(i)2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (5)$$

$$z(i+1) = z(i) - \rho(i)2^{-i} \quad (6)$$

Fig. 2 shows the proposed architecture for the modified scaling-free CORDIC arithmetic, in which, eight shifters, two CSAs, two CLAs, two latches, and four MUXs are used; the shifters and MUXs are to determine $\rho(i)$.

2.2 The modified scaling-free radix-8 CORDIC Algorithm

By using the modified angle recoding method [12]-[13], the input angle ψ can be divided as follows.

$$\psi = \sum_{i=p}^{w-1} \phi(i) \tan^{-1} 2^{-i} \quad (7)$$

where $\phi(i) \in \{0,1\}$, and w is the word length. The CORDIC iteration is therefore represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & \phi(i)2^{-i} \\ -\phi(i)2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (8)$$

$$z(i+1) = z(i) - \phi(i) \tan^{-1} 2^{-i} \quad (9)$$

Let $i = 3n - c$; $c \in \{0,1,2\}$. By using the Taylor series expansion, the absolute difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$ is given by

$$\zeta = \left| \tan^{-1} 2^{-(3n-c)} - 2^c \tan^{-1} 2^{-3n} \right| = \left| \frac{1}{3} \cdot 2^{-3(3n-c)} + \Lambda \right| \quad (10)$$

where Λ is the remaining terms of the difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$. Thus, we have

$$\zeta \cong \frac{2^{-3(3n-c)}}{3} = \frac{2^{-3i}}{3} \quad (11)$$

For w -bit operands, ζ can be ignored in the following sense

$$\zeta \leq 2^{-w} \quad (12)$$

Based on equations (11) and (12), we have

$$\frac{2^{-3i}}{3} \leq 2^{-w} \quad (13)$$

$$3i + \log_2 3 \geq w \quad (14)$$

$$i \geq \frac{w - \log_2 3}{3} = \left\lceil \frac{w - \log_2 3}{3} \right\rceil \cong \frac{w}{3} \quad (15)$$

As a result, when $i > \frac{w}{3}$, three consecutive terms of equation (7) can be integrated into a single term as follows:

$$\begin{aligned} & \phi(3n-2) \tan^{-1}(2^{-(3n-2)}) + \phi(3n-1) \tan^{-1}(2^{-(3n-1)}) \\ & + \phi(3n) \tan^{-1}(2^{-(3n)}) \\ & = (\phi(3n-2) \cdot 2^2 + \phi(3n-1) \cdot 2^1 + \phi(3n) \cdot 2^0) \tan^{-1}(2^{-3n}) \\ & = \varphi(n) \tan^{-1}(2^{-3n}) \end{aligned} \quad (16)$$

where $\phi(\cdot) \in \{0,1\}$, and therefore $\varphi(n) \in \{0,1,2,\dots,7\}$. It follows that the resulting radix-8 CORDIC algorithm is represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = K_8(i) \begin{bmatrix} 1 & -\varphi(i) \cdot 2^{-3i} \\ \varphi(i) \cdot 2^{-3i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (17)$$

$$z(i+1) = z(i) - \varphi(i) \tan^{-1} 2^{-3i} \quad (18)$$

$$K_8(i) = (1 + \varphi^2(i) 2^{-6i})^{1/2} \quad (19)$$

The scaling factor K_8 is given by

$$K_8 = \prod_{i=p}^{i=\lceil w/3 \rceil - 1} K_8(i) \quad (20)$$

It can be shown that the scaling factor turns out to be equal to 1 when the input angle is less than $2^{-w/2}$, and moreover, if the input angle is less than $2^{-w/3}$, equation (18) can be rewritten as [19]

$$z(i+1) = z(i) - \varphi(i) 2^{-3i} \quad (21)$$

Fig. 3 depicts the proposed architecture for the modified scaling-free radix-8 CORDIC arithmetic. In which, six shifters, two CSAs, two CLAs, and two latches are used; the shifters and switches are to determine the radices for computations. Note that the number of processors is reduced, and system throughput is increased at the cost of hardware complexity.

2.3 The proposed hybrid CORDIC Algorithm

The input angle Ω can be decomposed into a higher-angle Ω_H and a lower-angle Ω_L represented as

$$\Omega = \Omega_H + \Omega_L = \sum_{i=0}^{\lceil u/2 \rceil - 1} \rho(i) 2^{-2i} + \sum_{i=\lceil u/2 \rceil}^{\lceil u/2 \rceil + \lceil (w-u)/3 \rceil - 1} \varphi(i) 2^{u-1+3(i-\lceil w/4 \rceil)} \quad (22)$$

where w is the word length with the first u bits being the most significant bits; Ω_H and Ω_L are computed by using the modified scaling-free CORDIC algorithm and the modified scaling-free radix-8 CORDIC algorithm, respectively. For computation efficiency, the determination of u is as follows: 1) u must be an odd number to satisfy the MBE method, and 2) $u \geq \frac{w}{2}$ under the scaling-free constraint. Thus, we have

$$u = \begin{cases} 2n+1, & \text{if } w = 4n+0 \\ 2n+1, & \text{if } w = 4n+1 \\ 2n+1, & \text{if } w = 4n+2 \\ 2n+3, & \text{if } w = 4n+3 \end{cases} \quad (23)$$

Based on the above equation, the minimum iteration number of the proposed hybrid CORDIC algorithm can be obtained as shown in Fig. 4. The computations of $x(i)$ and $y(i)$ are therefore as follows.

$$\text{For } \frac{p}{2} \leq i < \left\lceil \frac{u}{2} \right\rceil,$$

$$x(i+1) = x(i) - \rho^2(i)2^{-(4i+1)}x(i) - \rho(i)2^{-2i}y(i) \quad (24)$$

$$y(i+1) = y(i) - \rho^2(i)2^{-(4i+1)}x(i) + \rho(i)2^{-2i}y(i) \quad (25)$$

$$\text{For } \left\lceil \frac{u}{2} \right\rceil \leq i < \left\lceil \frac{w}{4} \right\rceil + \left\lceil \frac{w-u}{3} \right\rceil,$$

$$x(i+1) = x(i) - \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}y(i) \quad (26)$$

$$y(i+1) = y(i) + \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}x(i) \quad (27)$$

3 Hardware Implementation of the Proposed DDFS

In this section, the DDFS implemented by using the hybrid CORDIC algorithm is presented. Fig. 5 shows the 16-bit DDFS architecture consisting mainly of phase accumulator, phase calculator, and sine/cosine generator, which is different from the conventional architecture. It is noted that the accumulated error in the sine/cosine generator is to be corrected by using the 4×16 -bit correction table. Take into account DAC technology, hardware cost and practical applications, the word length of the propose DDFS is set to 16-bit.

The hybrid CORDIC-based sine/cosine generator with recursively accumulated angle \mathcal{G}_m is given by

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} \cos \mathcal{G}_m & -\sin \mathcal{G}_m \\ \sin \mathcal{G}_m & \cos \mathcal{G}_m \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (28)$$

where $\mathcal{G}_m = \Delta_{acc} \frac{2\pi}{2^{16}}$, and Δ_{acc} is an integer number.

(29)

For convergence, the input angle of the scale-free CORDIC algorithm is restricted as follows:

$$\mathcal{G}_m < \sum_{i=4}^{w-1} 2^{-i} \cong \frac{1}{8} \quad (30)$$

From the above two equations, we have

$$\Delta_{acc} = \frac{2^{16}}{2\pi} \cdot \mathcal{G}_m < 1304 \quad (31)$$

The architecture for the sine/cosine generator is shown in Fig. 5. In which three modified scaling-free CORDIC arithmetic units (MCORDIC-Type A) and

two modified scaling-free radix-8 CORDIC arithmetic units (MCORDIC-Type B) are used. According to equation (23).

The chip is synthesized by the TSMC 0.18 μm 1P6M CMOS cell libraries [14]. The layout view of the proposed DDFS is shown in Figure 6. The core size obtained by the Synopsys[®] design analyzer is $612 \times 612 \mu\text{m}^2$. The power consumption obtained by the PrimePower[®] is 6.05 mW with a clock rate of 100MHz at 1.8V. The tuning latency is 8 clock cycles. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

4 Performance Analysis of the Proposed DDFS

The number of correcting points versus the SFDRs with different (F_s/F_o) 's in the proposed DDFS is shown in Fig. 7. Due to trade-off between hardware cost and system performance, the correcting circuit with 16 points is implemented in the proposed DDFS. Thus, the SFDR of the proposed DDFS is more than 84.4 dBc. Table 1 shows various comparisons of the proposed DDFS with other methods in [6] and [10]. As one can see, the proposed DDFS is superior in terms of SFDR, hardware cost, and power consumption.

5 Conclusion

The hybrid CORDIC-based multiplier-less DDFS architecture with small ROM and pipelined data path has been implemented. A SoC designed by 1P6M CMOS has been emulated on Xilinx XC2V6000 FPGA. For 16-bit DDFS, the SFDR of sine and cosine using the proposed architecture are more than 84.4 dBc. Simulation results show that the hybrid CORDIC-based approach is superior to the traditional approach to the design and implementation of DDFS, in terms of SFDR, power consumption, and hardware cost. The 16-bit DDFS is a reusable IP, which can be implemented in various processes with efficient uses of hardware resources for trade-offs of performance, area, and power consumption.

References:

- [1] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital frequency synthesis," *IEEE Proceedings of the Frequency Control Symposium*, June 5-7 1996, pp.942-950.

[2] S. C. Yi, K. T. Lee, J. J. Chen, C. H. Lin, "A low power efficient direct digital frequency synthesizer based on new two-level lookup table," *IEEE Canadian Conference on Electrical and Computer Engineering* 2006, May 2006, pp.963-966.

[3] D. A. Sunderland, R. A. Srauch, S. S. Wharfield, H. T. Peterson, C. R. Cole, "CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications," *IEEE Journal of Solid-State Circuits*, Vol.19, No.4, August 1984, pp.497-506.

[4] H. T. Nicholas, H. Samueli, B. Kim, "The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects," *IEEE 42nd Annual Frequency Control Symposium*, June 1-3 1988, pp.357-363.

[5] D. D. Caro, E. Napoli, A. G. M. Strollo, "Direct digital frequency synthesizers with polynomial hyperfolding technique," *IEEE Transactions of Circuits and Systems-II: Express Briefs*, Vol.51, No.7, July 2004, pp.337-344.

[6] D. Fu, A. N. Willson, Jr. "A high-speed processor for digital sine/cosine generation and angle rotation" in *Proc. 32nd Asilomar Conf. Signals, Systems and Computers*, Vol.1 1998, pp.177-181

[7] A. Torosyan, D. Fu, A. N. Willson, Jr, "A 300-MHz quadrature direct digital synthesizer/mixer in 0.25- μm CMOS" *IEEE Journal of Solid-State Circuits*, Volume 38, Issue 6, June 2003 pp. 875 - 887

[8] A. Madisetti, A. Y. Kwentus, A. N. Willson Jr, "A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range," *IEEE Journal of Solid-State Circuits*, Vol.34, No.8, August 1999, pp.1034-1043.

[9] E. Grayver, B. Daneshrad, "Direct digital synthesis using a modified CORDIC," *IEEE International Symposium on Circuits and Systems (ISCAS '98)*. Vol.5, May 1998, pp.241-244.

[10] C. Y. Kang, E. E. Swartzlander Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.53, No.5, May 2006, pp.1035-1044.

[11] T. Y. Sung, H. C. Hsin, "Design and simulation of reusable IP CORDIC core for special-purpose processors," *IET Computers & Digital Techniques*, Vol.1, No.5, Sept. 2007, pp.581-589.

[12] Y. H. Hu, S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Transactions on Computers*, Vol.42, No.1, January 1993, pp.99-102.

[13] T. B. Juang, S. F. Hsiao, M. Y. Tsai, "Para-CORDIC: parallel CORDIC rotation algorithm," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol.51, No.8, August 2004, pp.1515-1524.

[14] "TSMC 0.18 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.

Table I Comparison with previous works

CORDIC Based DDFS	Madisetti [8] 1999	Swartzlander [10] 2006	This work 2009
Process (μm)	1.0	0.13	0.18
Core Area (mm^2)	0.306	0.35	0.375
Maximum Sampling Rate (MHz)	80.4	1018	100
Power Consumption (mw)	40.602	350	6.056
Power Consumption (mw/MHz)	0.505	0.343	0.06
SFDR (dB_c)	81	90	84.4
Output Resolution (bits)	16	16	16
Tuning Latency (clock cycles)	16	--	8

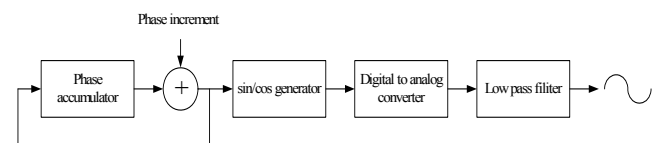


Fig. 1 The conventional DDFS architecture

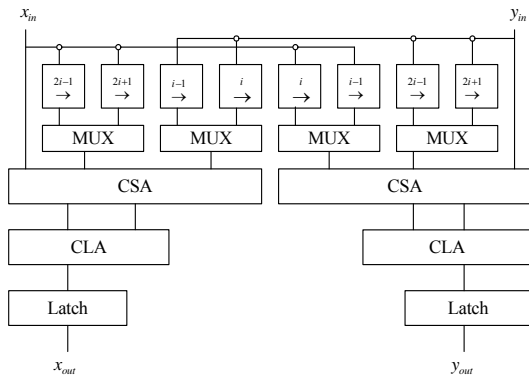


Fig. 2 The proposed architecture of modified scaling-free CORDIC arithmetic for computing θ_H (MCORDIC-Type A)

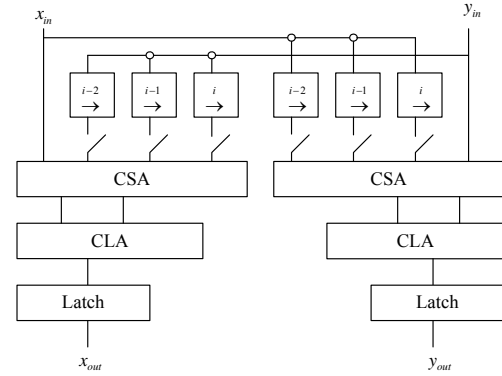


Fig. 3 The architecture of modified scaling-free radix-8 CORDIC arithmetic for computing θ_L (MCORDIC-Type B)

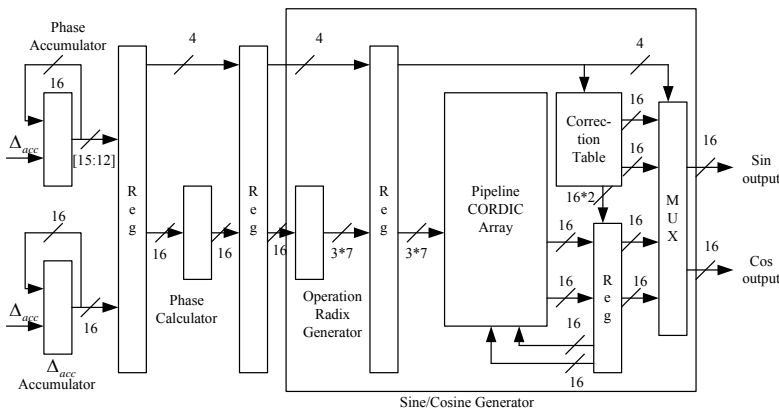


Fig. 4 The 16-bit DDFS architecture

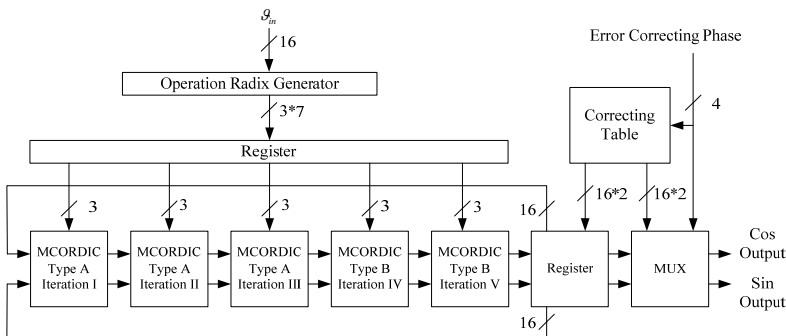


Fig. 5 The architecture of sine/cosine generator (The θ_{in} is an accumulated angle)

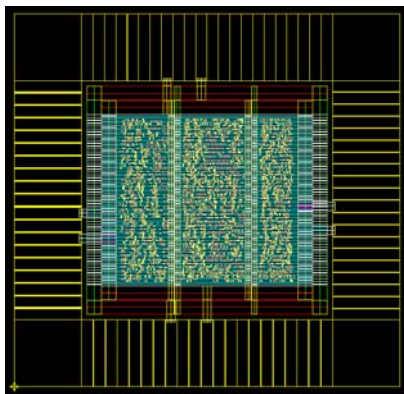


Fig. 6 The layout view of the proposed DDFS

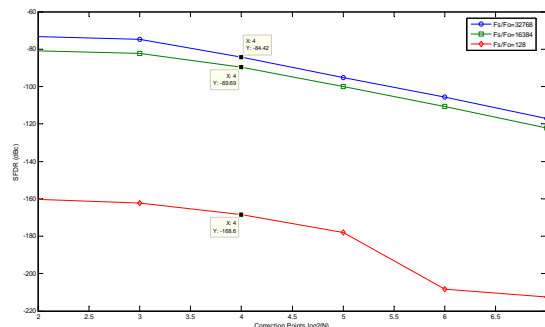


Fig. 7 Plot of the number of correcting points versus SFDRs with different (F_s / F_o) 's

Reconfigurable VLSI Architecture for FFT Processor

Tze-Yun Sung
 Department of Microelectronics
 Engineering
 Chung Hua University
 707, Sec. 2, Wufu Road
 Hsinchu City 300-12, Taiwan
 bobsung@chu.edu.tw

Hsi-Chin Hsin
 Department of Computer Science
 and Information Engineering
 National United University
 Miaoli 36003, Taiwan
 hsin@nuu.edu.tw

Abstract: - This paper presents a CORDIC (Coordinate Rotation Digital Computer)-based split-radix fast Fourier transform (FFT) core for OFDM systems, for example, Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL), and Worldwide Interoperability for Microwave Access (WiMAX). High-speed 128/256/512/1024/ 2048/4096/8192-point FFT processors have been implemented by 0.18 (1p6m) at 1.8V, in which all the control signals are generated internally. These FFT processors outperform the conventional ones in terms of both power consumption and core area.

Key-Words: - IP, FFT, split-radix, CORDIC, OFDM.

1 Introduction

High-performance fast Fourier transform (FFT) processor is needed especially for real-time digital signal processing applications. Specifically, the computation of discrete Fourier transform (DFT) ranging from 128 to 8192 points is required for the orthogonal frequency division multiplexer (OFDM) of the following standards: Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL) and Worldwide Interoperability for Microwave Access (WiMAX) [1]-[8]. Thompson [9] proposed an efficient VLSI architecture for FFT in 1983. Wold and Despain [10] proposed pipelined and parallel-pipelined FFT for VLSI implementations in 1984. Widhe [11] developed efficient processing elements of FFT in 1997. To reduce the computation complexity, the split-radix 2/4, 2/8, and 2/16 FFT algorithms were proposed in [12]-[15].

As the Booth multiplier is not suitable for hardware implementations of large FFT, we propose the CORDIC-based multiplier. Moreover, we develop a ROM-free twiddle factor generator using simple shifters and adders only [1], which obviates the need to store all the twiddle factors in a large ROM space. As a result, the proposed CORDIC-based split-radix FFT core with the ROM-free twiddle factor generator is suitable for the wireless local area network

(WLAN) applications. In this paper, a high-performance 128/256/512/1024/2048/4096/8192-point FFT processor is presented for the European and Japanese standards. The remainder of this paper proceeds as follows. In Section II, the split-radix 2/8 FFT algorithm and the CORDIC algorithm are reviewed briefly. In Section III, the reusable IP 128-point CORDIC-based split-radix FFT core is proposed. In Section IV, the hardware implementations of FFT processors are described. The performance analysis is presented in Section V. Finally, the conclusion is given in Section VI.

2 Review of Split-Radix FFT and CORDIC Algorithm

2.1 Split-Radix FFT

The idea behind the split-radix FFT algorithm is to compute the even and odd terms of FFT separately. The even term of the split-radix 2/8 FFT algorithm is given by

$$X(2k) = \sum_{n=0}^{N/2-1} (x(n) + x(n + \frac{N}{2}))W_{N/2}^{nk} \quad (1)$$

where $W_{N/2} = e^{-j\frac{2\pi}{N/2}}$ and $k = 0, 1, 2, \dots, (N/2) - 1$.

The odd term is as follows:

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\begin{aligned}
 X(8k+l) = & \sum_{n=0}^{N/8-1} \left((x(n) + x(n + \frac{2N}{8})W_4^l + x(n + \frac{4N}{8})W_4^{2l} \right. \\
 & + x(n + \frac{6N}{8})W_4^{3l} + x(n + \frac{N}{8}) + x(n + \frac{3N}{8})W_4^l \quad (2) \\
 & \left. + x(n + \frac{5N}{8})W_4^{2l} + x(n + \frac{7N}{8})W_4^{3l} \right) W_N^{nl} W_{N/8}^{nk}
 \end{aligned}$$

where $k = 0, 1, 2, \dots, (N/8) - 1$ and $l = 1, 3, 5, 7$. The split-radix 2/8 FFT algorithm, which combined with radix-2 and radix-4 proves effective to develop a reusable IP 128-point FFT core.

2.2 CORDIC Algorithm

The CORDIC algorithm in the circular coordinate system is as follows [16].

$$x(i+1) = x(i) - \sigma_i 2^{-i} y(i) \quad (3)$$

$$y(i+1) = y(i) + \sigma_i 2^{-j} x(i) \quad (4)$$

$$z(i+1) = z(i) - \sigma_i \alpha(i) \quad (5)$$

$$\alpha(i) = \tan^{-1} 2^{-i} \quad (6)$$

where $\sigma_i = \text{sign}(z(i))$ with $z(i) \rightarrow 0$ in the rotation mode, and $\sigma_i = -\text{sign}(x(i)) \cdot \text{sign}(y(i))$ with $y(i) \rightarrow 0$ in the vectoring mode. The scale factor:

$k(i)$ is equal to $\sqrt{1 + \sigma_i^2 2^{-2i}}$. After n micro-rotations, the product of the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k(i) = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \quad (7)$$

Notice that CORDIC in the circular coordinate system with rotation mode can be written by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = K_c \begin{bmatrix} \cos z_0 & \sin z_0 \\ -\sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (8)$$

where $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ and $\begin{bmatrix} x_n \\ y_n \end{bmatrix}$ are the input vector and the

output vector, respectively, z_0 is the rotation angle, and K_c is the scale factor. In [1], the circular rotation computation of CORDIC was used for complex multiplication with $e^{-j\theta}$, which is given by

$$\begin{bmatrix} \text{Re}[X'] \\ \text{Im}[X'] \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \text{Re}[X] \\ \text{Im}[X] \end{bmatrix} \quad (9)$$

3 Reusable IP 128-Point CORDIC-Based Split-Radix FFT Core

Figure 1 shows the proposed 128-point CORDIC-based split-radix FFT processor, which can be used as a reusable IP core for various FFT with multiples of 128 points. Notice that the modified split-radix 2/8 FFT butterfly processor and the

ROM-free twiddle factor generator are used. In addition, an internal (128 32-bit) SRAM is used to store the input and output data for hardware efficiency, through the use of the in-place computation algorithm [1]

3.1 CORDIC-Based Split-Radix 2/8 FFT Processor

For the butterfly computation of the proposed CORDIC-based split-radix 2/8 FFT processor, sixteen complex additions, two constant multiplications (CM), and four CORDIC operations are needed, as shown in Figure 2. The CORDIC algorithm has been widely used in various DSP applications because of the hardware simplicity. According to equation (9), the twiddle factor multiplication of FFT can be considered a 2-D vector rotation in the circular coordinate system. Thus, CORDIC in the circular coordinate system with rotation mode is adopted to compute complex multiplications of FFT.

The pipelined CORDIC arithmetic unit can be obtained by decomposing the CORDIC algorithm into a sequence of operational stages. In [17], we derived the error analysis of fixed-point CORDIC arithmetic, based on which, the number of the CORDIC stages can be determined effectively. For example, the number of the CORDIC stages is 12 if the overall relative error of 16-bit CORDIC arithmetic is required to be less than 10^{-3} . The pipelined CORDIC arithmetic unit with 12 stages and an additional pre-scalar stage. In which, the pre-calculated scaling factor $K_c \approx 1.64676$ and the Booth binary recoded format leads to 1.101001. The main concern for the design of the CORDIC arithmetic unit is throughput rather than latency. The proposed CORDIC arithmetic unit in terms of gate counts is less than 4 real multipliers significantly. In addition, the power consumption can be reduced significantly by using the proposed CORDIC arithmetic unit; it has been reduced by 30% according to the report of PrimePower® distributed by Synopsys.

As the twiddle factors: W_8^1 and W_8^3 are equal to $\frac{\sqrt{2}}{2}(1-j)$ and $-\frac{\sqrt{2}}{2}(1+j)$, respectively, a complex number, say $(a+bj)$, times W_8^1 or W_8^3 can be written by

$$(a+bj) \times \left(\frac{\sqrt{2}}{2}(1-j) \right) = \frac{\sqrt{2}}{2} ((a+b) + j(-a+b)) \quad (10)$$

$$(a+bj) \times \left(-\frac{\sqrt{2}}{2}(1+j) \right) = -\frac{\sqrt{2}}{2} ((a-b) + j(a+b)) \quad (11)$$

where $\frac{\sqrt{2}}{2}$ can be represented as 1.0101010 using the Booth binary recoded form (BBRF). Thus, the CM unit can be implemented by using simple adders and shifters only. Figure 3 shows the pipelined CM architecture, which uses three subtractions/additions and therefore improves on the computation speed significantly.

Based on the above-mentioned CORDIC arithmetic unit and CM unit, the computational circuit and hardware architecture of the CORDIC-based split-radix 2/8 FFT butterfly computation are realized. As one can see, the pipelined CORDIC arithmetic unit aims at increasing the throughput of complex multiplications..

3.2 ROM-Free Twiddle Factor Generator

In the conventional FFT processor, a large ROM space is needed to store all the twiddle factors. To reduce the chip area, a twiddle factor generator is thus proposed. Figure 4 shows the ROM-free twiddle factor generator using simple adders and shifters for 128-point FFT. In which, the 16-bit accumulator is to generate the value $2n\pi$ for each index n ; $n = 2^{\log_2 N - 3} - 1$, the 16-bit shifter is to divide $2n\pi$ by N , and the 16-bit shifter/adder is to produce the twiddle factors: θ_N^{1n} , θ_N^{3n} , θ_N^{5n} and θ_N^{7n} . By using the twiddle factor generator, the chip area and power consumption can be reduced significantly at the cost of an additional logic circuit. Table 1 shows the gate counts of the full-ROM storing all the twiddle factors, the CORDIC twiddle factor generator [1] and the ROM-free twiddle factor generator.

4 Implementation of FFT Processors

The 128/256/512/1024/2048/4096/8192-point FFT processors. In which, the radix-2 and split-radix 2/4 butterfly processors [1] using the pipelined CORDIC arithmetic units and twiddle factor generators are implemented; and moreover, two memory banks (4096/2048/1024/512/256/0 \times 32-bit and 8192/4096/2048/1024/512/256/128 \times 32-bit) are allocated for increased efficiency by using the in-place computation algorithm [1]. Hardware architecture is shown in Figure 5.

The hardware code written in Verilog[®] is running on a workstation with the ModelSim[®] simulation tool and Synopsys[®] synthesis tool (design compiler). The chips are synthesized by the TSMC 0.18 μm 1p6m CMOS cell libraries [18]. The physical circuit is synthesized by the Astro[®] tool. The circuits are evaluated by DRC, LVS and PVS [19].

The layout view of the 8192-point FFT processor is

shown in Figure 6. The core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point and 8192-point FFT processors are shown in Table 2. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

5 Performance Analysis of The Proposed FFT Architecture

FFT processors used to compute 128/256/512/1024/2048/4096/8192-point FFT are composed mainly of the 128-point CORDIC-based split-radix 2/8 FFT core; the computation complexity using a single 128-point FFT core is $O(N/6)$ for N -point FFT. The log-log plot of the CORDIC computations versus the number of FFT points is shown in Figure 7. As one can see, the proposed FFT architecture is able to improve the power consumption and computation speed significantly.

6 Conclusion

This paper presents low-power and high-speed FFT processors based on CORDIC and split-radix techniques for OFDM systems. The architectures are mainly based on a reusable IP 128-point CORDIC-based split-radix FFT core. The pipelined CORDIC arithmetic unit is used to compute the complex multiplications involved in FFT, and moreover the required twiddle factors are obtained by using the proposed ROM-free twiddle factor generator rather than storing them in a large ROM space.

The CORDIC-based 128/256/512/1024/2048/4096/8192-point FFT processors have been implemented by 0.18 μm CMOS, which take 395 μs , 176.8 μs , 77.9 μs , 33.6 μs , 14 μs , 5.5 μs and 1.88 μs to compute 8192-point, 4096-point, 2048-point, 1024-point, 512-point, 256-point and 128-point FFT, respectively.

The CORDIC-based FFT processors are designed by using the portable and reusable Verilog[®]. The 128-point FFT core is a reusable intellectual property (IP), which can be implemented in various processes and combined with an efficient use of hardware resources for the trade-offs of performance, area, and power consumption.

References:

- [1] T. Y. Sung, "Memory-efficient and high-speed split-radix FFT/IFFT processor based on

- pipelined CORDIC rotations,” *IEE Proc.-Vis. Image Signal Proc.*, Vol. 153, No. 4, Aug. 2006, pp.405-410.
- [2] J. C. Kuo, C. H. Wen, A. Y. Wu, “Implementation of a programmable 64/spl sim/2048-point FFT/IFFT processor for OFDM-based communication systems,” *Proceedings of the 2003 International Symposium on Circuits and Systems*, Volume 2, 25-28 May 2003 pp.II-121 - II-124.
- [3] L. Xiaojin, Z. Lai, C. J. Cui, “A low power and small area FFT processor for OFDM demodulator,” *IEEE Transactions on Consumer Electronics*, Volume 53, Issue 2, May 2007, pp. 274 – 277.
- [4] J. Lee, H. Lee, S. I. Cho, S. S. Choi, “A high-speed, low-complexity radix-216 FFT processor for MB-OFDM UWB systems,” *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp.
- [5] A. Cortes, I. Velez, J. F. Sevillano, A. Irizar, “An approach to simplify the design of IFFT/FFT cores for OFDM systems,” *IEEE Transactions on Consumer Electronics*, Volume 52, Issue 1, Feb. 2006, pp.26 – 32.
- [6] Y. H. Lee, T. H. Yu, K. K. Huang, A. Y. Wu, “Rapid IP design of variable-length cached-FFT processor for OFDM-based communication systems,” *IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 2006 pp.62-65.
- [7] C. L. Wey, W. C. Tang, S. Y. Lin, “Efficient memory-based FFT architectures for digital video broadcasting (DVB-T/H),” *2007 International Symposium on VLSI Design, Automation and Test*, 25-27 April 2007, pp.1-4.
- [8] Y. W. Lin, H. Y. Liu, C. Y. Lee, “A 1-GS/s FFT/IFFT processor for UWB applications,” *IEEE Journal of Solid-State Circuits*, Volume 40, Issue 8, Aug. 2005, pp.1726-1735.
- [9] C. D. Thompson, “Fourier transform in VLSI,” *IEEE Transactions on Computers*, Vol.32, No. 11, 1983, pp.1047-1057.
- [10] E. H. Wold, A. M. Despain, “Pipelined and parallel-pipelined FFT processor for VLSI implementation,” *IEEE Transactions on Computers*, Vol.33, No. 5, 1984, pp.414-426.
- [11] T. Widhe, “Efficient implementation of FFT processing elements,” *Linkoping Studies in Science and Technology*, Thesis No. 619, Linkoping University, Sweden, 1997.
- [12] P. Duhamel, H. Hollmann, “Implementation of "split-radix" FFT algorithms for complex, real, and real symmetric data.” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Volume 10, April 1985, pp.784 – 787.
- [13] A .A. Petrovsky, S. L. Shkredov, “Automatic generation of split-radix 2-4 parallel-pipeline FFT processors: hardware reconfiguration and core optimizations,” *2006 International Symposium on Parallel Computing in Electrical Engineering*, pp.181-186.
- [14] S. Bouguezal, M. O. Ahmad, M. N. S. Swamy, “A new radix-2/8 FFT algorithm for length-q/spl times/2/sup m/ DFTs,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume 51, Issue 9, 2004, pp.1723- 1732.
- [15] W. C. Yeh, C. W. Jen, “High-speed and low-power split-radix FFT.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 51, Issue 3, March 2003, pp.864 – 874.
- [16] M. D. Ercegovac, T. Lang, “CORDIC algorithm and implementations.” Digital Arithmetic, Morgan Kaufmann Publishers, 2004, Chapter 11.
- [17] T. Y. Sung, H. C. Hsin, “Fixed-point error analysis of CORDIC arithmetic for special-purpose signal processors,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E90-A, No.9, Sep. 2007, pp.2006-2013.
- [18] “TSMC 0.18 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.
- [19] Cadence design systems: [http://www.cadence.com/products /pages/default.aspx](http://www.cadence.com/products/pages/default.aspx).

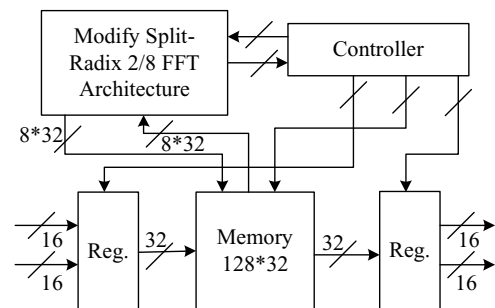


Fig. 1 The proposed 128-point CORDIC- based split-radix FFT processor

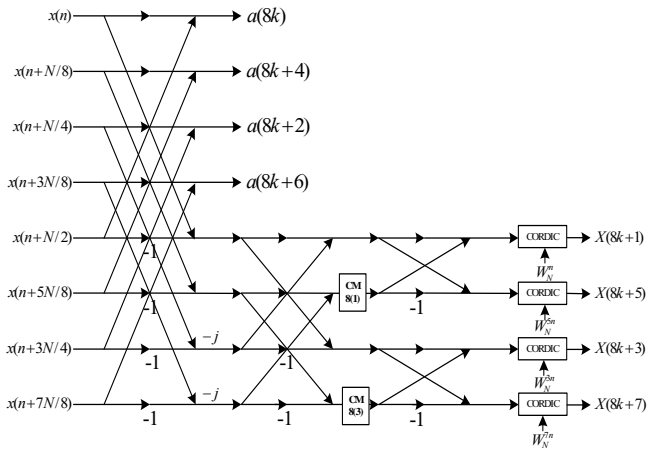


Fig. 2 Data flow of the butterfly computation of the modified split-radix 2/8 FFT

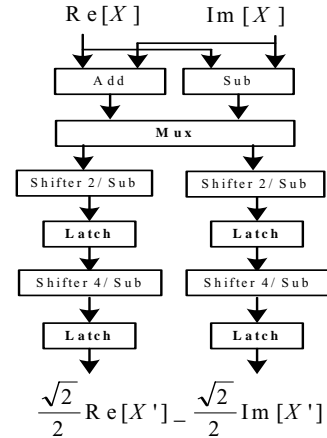


Fig. 3 Constant multiplier (CM) architecture for the modified split-radix 2/8 FFT

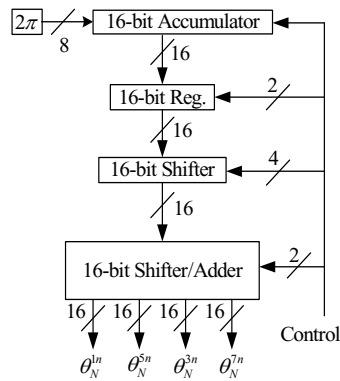


Fig. 4 Proposed ROM-free twiddle factor generator for 128-point FFT

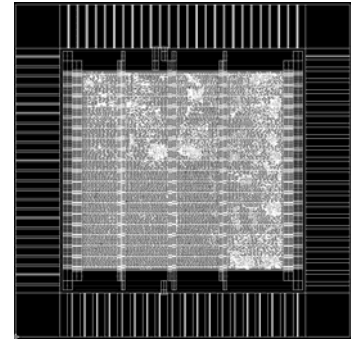


Fig. 6 Layout view of the 8192-point FFT processor

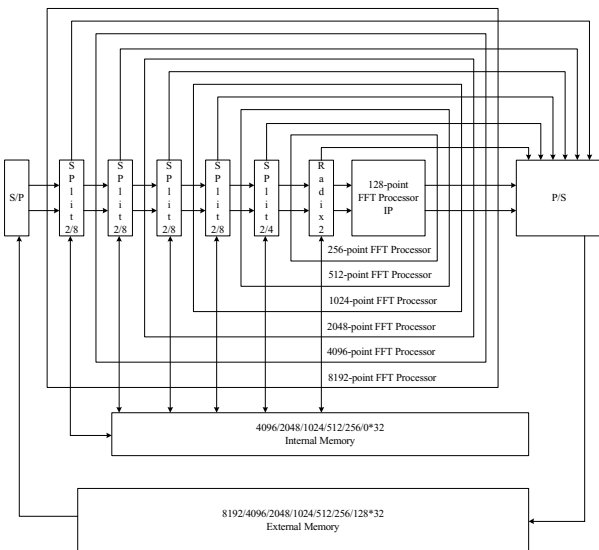


Fig. 5 Hardware architecture of the 128/256/512/1024/2048/4096/8192-point FFT processor

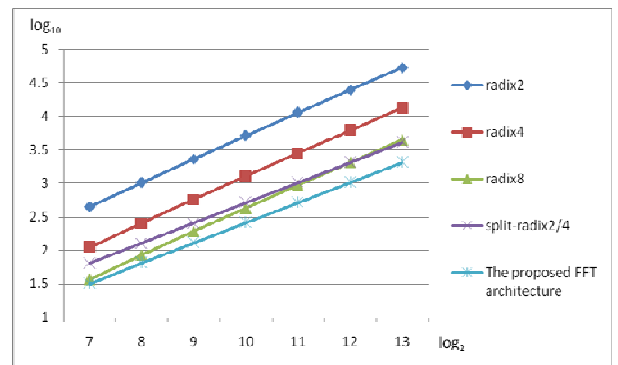


Fig. 7 Log-log plot of the CORDIC computations versus the number of FFT points

Table 1 Hardware requirements of the full-ROM, the CORDIC twiddle factor generator [1], and the ROM-free twiddle factor generator

Full-Twiddle Factor ROM					1bit~1gate
8192-Point ROM					
4K×16 bit					
CORDIC Twiddle Factor Generator (T. Y. Sung, 2006) [1]					
16-bit CORDIC ~18K bit	11-bit Adder ~150 gates	11-bit Shifter ~50 gates	16-bit Shifter ~90 gates	16-bit Adder ~200 gates	
ROM-free Twiddle Factor Generator (Sung, Hsin and Cheng, 2008)					
16-bit Accumulator ~200gates	16-bit Register ~32 gates	16-bit Shifter ~90 gates	16-bit Shifter/Adder ~90×2+200×2 gates		

Table 2 Core areas, power consumptions, clock rates of 128-, 256-, 512-, 1024-, 2048-, 4096- and 8192-point FFT generator

FFT Size	Core Area	Power Consumption	Clock Rate
128-point	2.28mm ²	80mW	200MHz
256-point	2.37mm ²	84mW	200MHz
512-point	2.49mm ²	88mW	200MHz
1024-point	2.62mm ²	94mW	200MHz
2048-point	2.81mm ²	99mW	200MHz
4096-point	3.10mm ²	106mW	200MHz
8192-point	3.62mm ²	117mW	200MHz

附件二：出席國際學術會議心得報告及發表之論文

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

98 年 05 月 26 日

報告人姓名	宋志雲	服務機構 及職稱	中華大學微電子工程學系 副教授
時間 會議 地點	98.05.20~98.05.22 中國 杭州市	本會核定 補助文號	計畫編號: NSC 97-2221-E-216-044
會議 名稱	(英文) 9 th WSEAS International Conference on Multimedia Systems & Signal Processing 8 th WSEAS International Conference on Instrumentation, Measurement, Circuit and Systems		
發表 論文 題目	(英文) High-SFDR and Multiplierless Direct Digital Frequency Synthesizer, A Simple Rate Distortion Estimation for Embedded Block Coding, Reconfigurable VLSI Architecture for FFT processor.		

報告內容應包括下列各項：

一、參加會議經過

發表論文三篇，分別發表於 9th WSEAS International Conference on Multimedia Systems & Signal Processing, 8th WSEAS International Conference on Instrumentation, Measurement, Circuit and Systems 並與與會人士討論及聽取議程主席之意見。本研討會為 WSEAS 之重要研討會，歷史悠久，至少有二十年以上之歷史。本人發表之論文接獲選為最佳論文，並邀請增加篇幅在其期刊(EI)登出。本人發表之論文，引起大家極大興趣，會中廣泛討論內容。

二、與會心得

此一研討會為 WSEAS 重要研討會，與會人士均為一時之選。會中可認識相關領域的領導人物，機會難得。論文如獲最佳論文，延伸後將獲得該學會之期刊(EI)刊登。

三、考察參觀活動(無是項活動者省略)

無

四、建議

鼓勵在教學之學校教師投稿此一研討會，因為有指標性意義。同時爭取 WSEAS 類似權威性研討會之主辦權，以增加台灣之國際學術地位。

五、攜回資料名稱及內容

研討會之論文光碟及紙本論文集。

六、其他

High-SFDR and Multiplierless Direct Digital Frequency Synthesizer

Tze-Yun Sung

Department of
Microelectronics Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
bobsung@chu.edu.tw

Hsi-Chin Hsin

Department of Computer
Science and Information
Engineering
National United University
Miaoli 36003, Taiwan
hsin@nuu.edu.tw

Lyu-Ting Ko

Department of Electrical
Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
m09601049@chu.edu.tw

Abstract: - This paper presents a hybrid CORDIC (COordinate Rotation DIgital Computer) algorithm for designs and implementations of the direct digital frequency synthesizer (DDFS). The proposed multiplier-less architecture with small ROM (4×16-bit) and pipelined data path provides a spurious free dynamic range (SFDR) of more than 84.4 dBc. A SoC (system on chip) has been designed by 1P6M CMOS, and then emulated on the Xilinx FPGA. It is shown that the hybrid CORDIC-based architecture is suitable for VLSI implementations of the DDFS in terms of hardware cost, power consumption, and SFDR.

Key-Words: - DDFS, hybrid CORDIC, SoC, FPGA, SFDR.

1 Introduction

The direct digital frequency synthesizer (DDFS) plays a key role in many digital communication systems. Fig. 1 depicts the conventional DDFS, which consists mainly of phase accumulator, sine/cosine generator, digital-to-analog converter, and low-pass filter. The sine/cosine generator as the core of DDFS is usually implemented by using a ROM lookup table; with high spurious free dynamic ranges (SFDR) comes a large ROM lookup table [1]. In order to reduce the size of the lookup table, many techniques were proposed [1]-[4]. The quadrant compression technique can reduce the ROM size by 75% [2]. The Sunderland architecture is to split the ROM into two smaller ones [3], and its improved version known as the Nicholas architecture results in a higher ROM-compression ratio (32:1) [4]. In [5], the polynomial hyperfolding technique with high order polynomial approximation was used to design DDFS. In [6] and [7], the angle rotation algorithm was used to design quadrature direct digital frequency synthesizer/complex mixer (QDDSM). COordinate Rotation DIgital Computer (CORDIC) is a well known arithmetic algorithm, which evaluates various elementary functions including sine and cosine functions by using simple adders and shifters only. Thus, CORDIC is suitable for the design of high-performance chips with VLSI technologies. Recently, the CORDIC algorithm has received a lot of attention to the design of high-performance DDFS [8]-[11], especially for the modern digital

communication systems.

This paper is organized as follows. In section II, the hybrid CORDIC algorithm is proposed. In section III, hardware implementation of DDFS is described. The performance analysis is presented in section IV. Finally, the conclusion is given in section V.

2 The Hybrid CORDIC Algorithm

In this section, the hybrid CORDIC algorithm is proposed, and based on which, a low-power and high-SFDR DDFS can be developed.

2.1 Modified Angle Recoding Method for CORDIC Algorithm

In order to reduce the number of CORDIC iterations, the input angle can be divided into encoded angles by using the modified Booth encoding (MBE) method [12]. Specifically, let ψ denote the input angle represented by

$$\psi = f(0)2^{-p} + f(1)2^{-(p+1)} + \dots + f(w-1)2^{-(w-1)} \quad (1)$$

where $f(i) \in \{0,1\}$, w is the word length of operands, and $\left\lceil \frac{(w-2.585)}{3} \right\rceil = p \leq i \leq w-1$. The MBE decomposition of ψ is as follows.

$$\psi = \sum_{i=p/2}^{(w-1)/2} \beta(i) \quad (2)$$

where the encoded angle:

$$\beta(i) = \rho(i)2^{-i} \text{ with } \rho(i) \in \{-2,-1,0,1,2\}. \text{ As } \sin \beta(i)$$

and $\cos \beta(i)$ can be approximated by

$$\sin \beta(i) \cong \beta(i) = \rho(i)2^{-i} \quad (3)$$

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\cos \beta(i) \cong 1 - \frac{\beta^2(i)}{2} = 1 - \rho^2(i)2^{-(2i+1)} \quad (4)$$

we have

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 - \rho^2(i)2^{-(2i+1)} & \rho(i)2^{-i} \\ -\rho(i)2^{-i} & 1 - \rho^2(i)2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (5)$$

$$z(i+1) = z(i) - \rho(i)2^{-i} \quad (6)$$

Fig. 2 shows the proposed architecture for the modified scaling-free CORDIC arithmetic, in which, eight shifters, two CSAs, two CLAs, two latches, and four MUXs are used; the shifters and MUXs are to determine $\rho(i)$.

2.2 The modified scaling-free radix-8 CORDIC Algorithm

By using the modified angle recoding method [12]-[13], the input angle ψ can be divided as follows.

$$\psi = \sum_{i=p}^{w-1} \phi(i) \tan^{-1} 2^{-i} \quad (7)$$

where $\phi(i) \in \{0,1\}$, and w is the word length. The CORDIC iteration is therefore represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & \phi(i)2^{-i} \\ -\phi(i)2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (8)$$

$$z(i+1) = z(i) - \phi(i) \tan^{-1} 2^{-i} \quad (9)$$

Let $i = 3n - c$; $c \in \{0,1,2\}$. By using the Taylor series expansion, the absolute difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$ is given by

$$\varsigma = \left| \tan^{-1} 2^{-(3n-c)} - 2^c \tan^{-1} 2^{-3n} \right| = \left| \frac{1}{3} \cdot 2^{-3(3n-c)} + \Lambda \right| \quad (10)$$

where Λ is the remaining terms of the difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$. Thus, we have

$$\varsigma \cong \frac{2^{-3(3n-c)}}{3} = \frac{2^{-3i}}{3} \quad (11)$$

For w -bit operands, ς can be ignored in the following sense

$$\varsigma \leq 2^{-w} \quad (12)$$

Based on equations (11) and (12), we have

$$\frac{2^{-3i}}{3} \leq 2^{-w} \quad (13)$$

$$3i + \log_2 3 \geq w \quad (14)$$

$$i \geq \frac{w - \log_2 3}{3} = \left\lceil \frac{w - \log_2 3}{3} \right\rceil \cong \frac{w}{3} \quad (15)$$

As a result, when $i > \frac{w}{3}$, three consecutive terms of equation (7) can be integrated into a single term as follows:

$$\begin{aligned} & \phi(3n-2) \tan^{-1}(2^{-(3n-2)}) + \phi(3n-1) \tan^{-1}(2^{-(3n-1)}) \\ & + \phi(3n) \tan^{-1}(2^{-(3n)}) \\ & = (\phi(3n-2) \cdot 2^2 + \phi(3n-1) \cdot 2^1 + \phi(3n) \cdot 2^0) \tan^{-1}(2^{-3n}) \\ & = \varphi(n) \tan^{-1}(2^{-3n}) \end{aligned} \quad (16)$$

where $\phi(\cdot) \in \{0,1\}$, and therefore $\varphi(n) \in \{0,1,2,\dots,7\}$. It follows that the resulting radix-8 CORDIC algorithm is represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = K_8(i) \begin{bmatrix} 1 & -\varphi(i) \cdot 2^{-3i} \\ \varphi(i) \cdot 2^{-3i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (17)$$

$$z(i+1) = z(i) - \varphi(i) \tan^{-1} 2^{-3i} \quad (18)$$

$$K_8(i) = (1 + \varphi^2(i) 2^{-6i})^{1/2} \quad (19)$$

The scaling factor K_8 is given by

$$K_8 = \prod_{i=p}^{\lceil w/3 \rceil - 1} K_8(i) \quad (20)$$

It can be shown that the scaling factor turns out to be equal to 1 when the input angle is less than $2^{-w/2}$, and moreover, if the input angle is less than $2^{-w/3}$, equation (18) can be rewritten as [19]

$$z(i+1) = z(i) - \varphi(i) 2^{-3i} \quad (21)$$

Fig. 3 depicts the proposed architecture for the modified scaling-free radix-8 CORDIC arithmetic. In which, six shifters, two CSAs, two CLAs, and two latches are used; the shifters and switches are to determine the radices for computations. Note that the number of processors is reduced, and system throughput is increased at the cost of hardware complexity.

2.3 The proposed hybrid CORDIC Algorithm

The input angle Ω can be decomposed into a higher-angle Ω_H and a lower-angle Ω_L represented as

$$\Omega = \Omega_H + \Omega_L = \sum_{i=0}^{\lceil u/2 \rceil - 1} \rho(i) 2^{-2i} + \sum_{i=\lceil u/2 \rceil}^{\lceil u/2 \rceil + \lceil (w-u)/3 \rceil - 1} \varphi(i) 2^{u-1+3(i-\lceil w/4 \rceil)} \quad (22)$$

where w is the word length with the first u bits being the most significant bits; Ω_H and Ω_L are computed by using the modified scaling-free CORDIC algorithm and the modified scaling-free radix-8 CORDIC algorithm, respectively. For computation efficiency, the determination of u is as follows: 1) u must be an odd number to satisfy the MBE method, and 2) $u \geq \frac{w}{2}$ under the scaling-free constraint. Thus, we have

$$u = \begin{cases} 2n+1, & \text{if } w = 4n+0 \\ 2n+1, & \text{if } w = 4n+1 \\ 2n+1, & \text{if } w = 4n+2 \\ 2n+3, & \text{if } w = 4n+3 \end{cases} \quad (23)$$

Based on the above equation, the minimum iteration number of the proposed hybrid CORDIC algorithm can be obtained as shown in Fig. 4. The computations of $x(i)$ and $y(i)$ are therefore as follows.

$$\text{For } \frac{p}{2} \leq i < \left\lceil \frac{u}{2} \right\rceil,$$

$$x(i+1) = x(i) - \rho^2(i)2^{-(4i+1)}x(i) - \rho(i)2^{-2i}y(i) \quad (24)$$

$$y(i+1) = y(i) - \rho^2(i)2^{-(4i+1)}x(i) + \rho(i)2^{-2i}y(i) \quad (25)$$

$$\text{For } \left\lceil \frac{u}{2} \right\rceil \leq i < \left\lceil \frac{w}{4} \right\rceil + \left\lceil \frac{w-u}{3} \right\rceil,$$

$$x(i+1) = x(i) - \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}y(i) \quad (26)$$

$$y(i+1) = y(i) + \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}x(i) \quad (27)$$

3 Hardware Implementation of the Proposed DDFS

In this section, the DDFS implemented by using the hybrid CORDIC algorithm is presented. Fig. 5 shows the 16-bit DDFS architecture consisting mainly of phase accumulator, phase calculator, and sine/cosine generator, which is different from the conventional architecture. It is noted that the accumulated error in the sine/cosine generator is to be corrected by using the 4×16 -bit correction table. Take into account DAC technology, hardware cost and practical applications, the word length of the propose DDFS is set to 16-bit.

The hybrid CORDIC-based sine/cosine generator with recursively accumulated angle \mathcal{G}_m is given by

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} \cos \mathcal{G}_m & -\sin \mathcal{G}_m \\ \sin \mathcal{G}_m & \cos \mathcal{G}_m \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (28)$$

where $\mathcal{G}_m = \Delta_{acc} \frac{2\pi}{2^{16}}$, and Δ_{acc} is an integer number.

(29)

For convergence, the input angle of the scale-free CORDIC algorithm is restricted as follows:

$$\mathcal{G}_m < \sum_{i=4}^{w-1} 2^{-i} \cong \frac{1}{8} \quad (30)$$

From the above two equations, we have

$$\Delta_{acc} = \frac{2^{16}}{2\pi} \cdot \mathcal{G}_m < 1304 \quad (31)$$

The architecture for the sine/cosine generator is shown in Fig. 5. In which three modified scaling-free CORDIC arithmetic units (MCORDIC-Type A) and

two modified scaling-free radix-8 CORDIC arithmetic units (MCORDIC-Type B) are used. According to equation (23).

The chip is synthesized by the TSMC 0.18 μm 1P6M CMOS cell libraries [14]. The layout view of the proposed DDFS is shown in Figure 6. The core size obtained by the Synopsys[®] design analyzer is $612 \times 612 \mu\text{m}^2$. The power consumption obtained by the PrimePower[®] is 6.05 mW with a clock rate of 100MHz at 1.8V. The tuning latency is 8 clock cycles. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

4 Performance Analysis of the Proposed DDFS

The number of correcting points versus the SFDRs with different (F_s/F_o) 's in the proposed DDFS is shown in Fig. 7. Due to trade-off between hardware cost and system performance, the correcting circuit with 16 points is implemented in the proposed DDFS. Thus, the SFDR of the proposed DDFS is more than 84.4 dBc. Table 1 shows various comparisons of the proposed DDFS with other methods in [6] and [10]. As one can see, the proposed DDFS is superior in terms of SFDR, hardware cost, and power consumption.

5 Conclusion

The hybrid CORDIC-based multiplier-less DDFS architecture with small ROM and pipelined data path has been implemented. A SoC designed by 1P6M CMOS has been emulated on Xilinx XC2V6000 FPGA. For 16-bit DDFS, the SFDR of sine and cosine using the proposed architecture are more than 84.4 dBc. Simulation results show that the hybrid CORDIC-based approach is superior to the traditional approach to the design and implementation of DDFS, in terms of SFDR, power consumption, and hardware cost. The 16-bit DDFS is a reusable IP, which can be implemented in various processes with efficient uses of hardware resources for trade-offs of performance, area, and power consumption.

References:

- [1] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital frequency synthesis," *IEEE Proceedings of the Frequency Control Symposium*, June 5-7 1996, pp.942-950.

[2] S. C. Yi, K. T. Lee, J. J. Chen, C. H. Lin, "A low power efficient direct digital frequency synthesizer based on new two-level lookup table," *IEEE Canadian Conference on Electrical and Computer Engineering* 2006, May 2006, pp.963-966.

[3] D. A. Sunderland, R. A. Srauch, S. S. Wharfield, H. T. Peterson, C. R. Cole, "CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications," *IEEE Journal of Solid-State Circuits*, Vol.19, No.4, August 1984, pp.497-506.

[4] H. T. Nicholas, H. Samueli, B. Kim, "The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects," *IEEE 42nd Annual Frequency Control Symposium*, June 1-3 1988, pp.357-363.

[5] D. D. Caro, E. Napoli, A. G. M. Strollo, "Direct digital frequency synthesizers with polynomial hyperfolding technique," *IEEE Transactions of Circuits and Systems-II: Express Briefs*, Vol.51, No.7, July 2004, pp.337-344.

[6] D. Fu, A. N. Willson, Jr. "A high-speed processor for digital sine/cosine generation and angle rotation" in *Proc. 32nd Asilomar Conf. Signals, Systems and Computers*, Vol.1 1998, pp.177-181

[7] A. Torosyan, D. Fu, A. N. Willson, Jr, "A 300-MHz quadrature direct digital synthesizer/mixer in 0.25- μm CMOS" *IEEE Journal of Solid-State Circuits*, Volume 38, Issue 6, June 2003 pp. 875 - 887

[8] A. Madiseti, A. Y. Kwentus, A. N. Willson Jr, "A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range," *IEEE Journal of Solid-State Circuits*, Vol.34, No.8, August 1999, pp.1034-1043.

[9] E. Grayver, B. Daneshrad, "Direct digital synthesis using a modified CORDIC," *IEEE International Symposium on Circuits and Systems (ISCAS '98)*. Vol.5, May 1998, pp.241-244.

[10] C. Y. Kang, E. E. Swartzlander Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.53, No.5, May 2006, pp.1035-1044.

[11] T. Y. Sung, H. C. Hsin, "Design and simulation of reusable IP CORDIC core for special-purpose processors," *IET Computers & Digital Techniques*, Vol.1, No.5, Sept. 2007, pp.581-589.

[12] Y. H. Hu, S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Transactions on Computers*, Vol.42, No.1, January 1993, pp.99-102.

[13] T. B. Juang, S. F. Hsiao, M. Y. Tsai, "Para-CORDIC: parallel CORDIC rotation algorithm," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol.51, No.8, August 2004, pp.1515-1524.

[14] "TSMC 0.18 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.

Table I Comparison with previous works

CORDIC Based DDFS	Madiseti [8] 1999	Swartzlander [10] 2006	This work 2009
Process (μm)	1.0	0.13	0.18
Core Area (mm^2)	0.306	0.35	0.375
Maximum Sampling Rate (MHz)	80.4	1018	100
Power Consumption (mw)	40.602	350	6.056
Power Consumption (mw/MHz)	0.505	0.343	0.06
SFDR (dB_c)	81	90	84.4
Output Resolution (bits)	16	16	16
Tuning Latency (clock cycles)	16	--	8

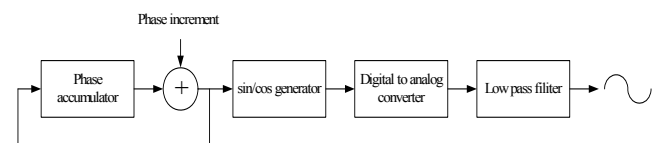


Fig. 1 The conventional DDFS architecture

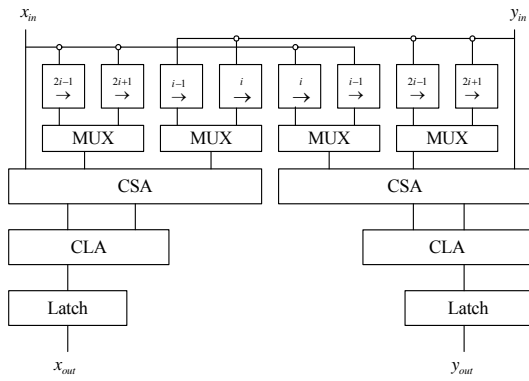


Fig. 2 The proposed architecture of modified scaling-free CORDIC arithmetic for computing θ_H (MCORDIC-Type A)

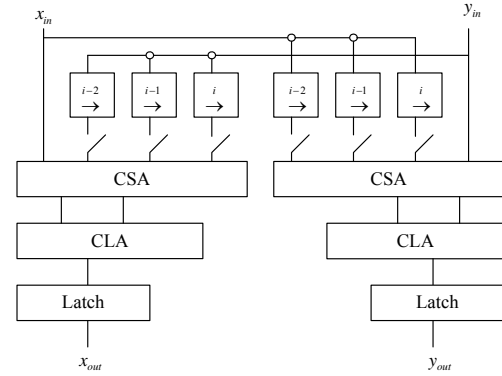


Fig. 3 The architecture of modified scaling-free radix-8 CORDIC arithmetic for computing θ_L (MCORDIC-Type B)

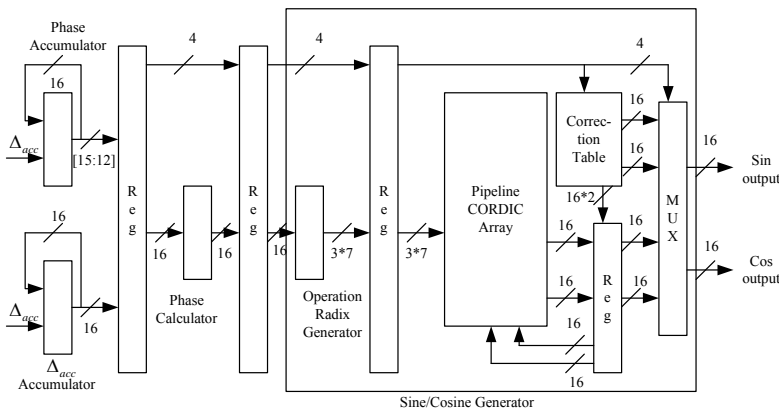


Fig. 4 The 16-bit DDFS architecture

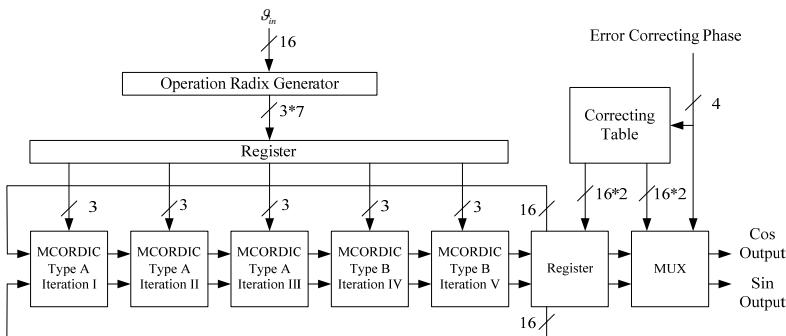


Fig. 5 The architecture of sine/cosine generator (The θ_{in} is an accumulated angle)

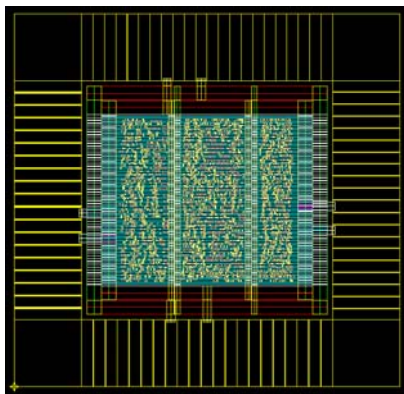


Fig. 6 The layout view of the proposed DDFS

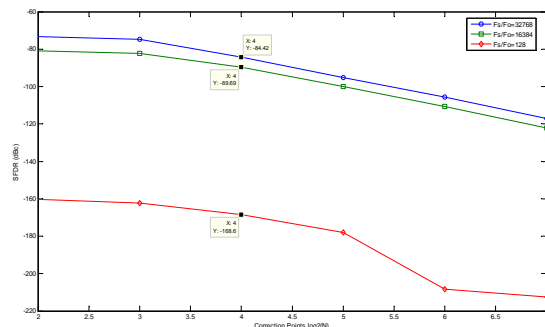


Fig. 7 Plot of the number of correcting points versus SFDRs with different (F_s / F_o) 's

A Simple Rate Distortion Estimation for Embedded Block Coding

HSI-CHIN HSIN

Department of Computer Science and
Information Engineering
National United University
1, Lien-Da, Miao-Li, 36003, Taiwan
hsin@nuu.edu.tw

TZE-YUN SUNG

Department of Microelectronics
Engineering
Chung Hua University
707 Wu-Fu, Hsinchu 30012, Taiwan
bobsung@chu.edu.tw

Abstract: - The embedded block coding with optimized truncation (EBCOT) algorithm has been adopted by the JPEG2000 standard. In which, the post compression rate distortion (PCRD) optimization needs a large memory space to store all the code streams of code blocks; however, code blocks of less importance might not be used for the optimal decoded image at a bit rate. To avoid waste of computational power and memory space, a simple context based rate distortion estimation (CBRDE) is proposed to arrange the scanning order of code blocks in an adaptive manner. CBRDE is based on the MQ table, which is available at both encoder and decoder. Thus, there is no need to store and transmit the rate distortion information of code blocks. Experimental results show that the rate distortion curves are almost convex; this demonstrates the benefit of the proposed CBRDE.

Key-Words: - image coding; wavelet transform; JPEG2000; EBCOT; CBRDE

1 Introduction

Wavelet transform offers many desirable properties, e.g. progressive transmission and embedded coding, which are suitable for image coding [1]-[3]. In general, embedded code streams of an image are organized in decreasing order of information importance. The embedded block coding with optimized truncation (EBCOT) algorithm has been adopted by JPEG2000 [4], which is preferable to the JPEG standard at the cost of increasing complexity [5]. For real-time applications, dedicated hardware is required to implement JPEG2000.

In EBCOT, the tier-1 algorithm applies embedded block coding with an arithmetic coder to each code block of the transform coefficients; the tier-2 algorithm performs rate distortion control by the post compression rate distortion (PCRD) algorithm. One of the crucial implementation issues of EBCOT is the design of PCRD, which requires a large amount of memory space to store all the code streams of code blocks with their respective rate distortion information. As all the code blocks of an image must be processed before PCRD, however, some code blocks may not be needed to reconstruct the optimal decoded image at a bit rate; this surely leads to waste of computational power. In [6], Fang et al. proposed a precompression rate distortion optimization algorithm to reduce the memory space by ignoring the unused code streams.

In this paper, we propose a context based rate distortion estimation (CBRDE) to arrange the scanning order of code blocks so that the available coding bits can be allocated for the most significant

code block. The proposed CBRDE is based on the MQ table of EBCOT, which is available at both encoder and decoder. As a result, the scanning order of code blocks can be also obtained at decoder, and therefore transmission of the contributions of each code block can be omitted. The remainder of this paper proceeds as follows. In Section 2, wavelet transform and EBCOT are reviewed briefly. Section 3 presents the proposed CBRDE algorithm to arrange the code blocks of an image in an adaptive manner. Experimental results are given in Section 4, and conclusion can be found in Section 5.

2 Wavelet Transform and EBCOT

Wavelet transform provides many desirable properties such as multiresolution analysis, subband decomposition with orientation selectivity, joint space-spatial frequency localization, self similarity across subbands of the same orientation, and energy clustering within each subband. In this section, wavelet transform and the EBCOT algorithm are reviewed briefly.

2.1 Wavelet transform

The wavelet transform of a signal, $S_\ell(n)$, at resolution ℓ is as follows.

$$\begin{aligned} S_{\ell+1}(n) &= \sum_k S_\ell(k) h(2n-k) \\ D_{\ell+1}(n) &= \sum_k S_\ell(k) g(2n-k) \end{aligned} \quad (1)$$

where $S_{\ell+1}(n)$ is the approximation at the next coarser resolution $\ell+1$, $D_{\ell+1}(n)$ is the detail between resolutions ℓ and $\ell+1$, $h(n) = \langle \phi, \phi_{-1,-n} \rangle$, $g(n) = \langle \psi, \phi_{-1,-n} \rangle$, $\langle \cdot, \cdot \rangle$ is an inner product operator, ψ is a valid (mother) wavelet, ϕ is the corresponding scaling function and $\phi_{-1,-n}(x) = 2^{-1/2} \phi(2^{-1}x - n)$. $S_{\ell}(n)$ can be exactly reconstructed from $S_{\ell+1}(n)$ and $D_{\ell+1}(n)$ by using following inverse wavelet transform.

$$S_{\ell}(n) = \sum_k S_{\ell+1}(k) \tilde{h}(n-2k) + \sum_k D_{\ell+1}(k) \tilde{g}(n-2k) \quad (2)$$

where $\tilde{h}(n) = h(-n)$ and $\tilde{g}(n) = g(-n)$.

For image applications, 2-D wavelet transform can be obtained by the tensor product of two 1-D wavelet transforms. Similarly, 2-D inverse wavelet transform can be obtained by the tensor product of two 1-D inverse wavelet transforms.

2.2 The EBCOT algorithm

In many cases, the DWT-based JPEG2000 standard outperforms the DCT-based JPEG standard. The idea behind the heart of JPEG2000 known as the EBCOT algorithm is to exploit energy clustering of wavelet coefficients. EBCOT is a two-tier algorithm; tier-1 performs bit-plane coding (BPC) followed by arithmetic coding (AC); tier-2 aims for post compression rate distortion optimization. The quantized wavelet coefficients of an image are coded by a context-based arithmetic coder known as the MQ coder, in which the probability models are stored in the MQ table. Figure 1 depicts block diagram of the JPEG2000 encoder.

In EBCOT, each BPC is further divided into three coding passes, namely significance propagation pass, magnitude refinement pass, and cleanup pass. Four primitive coding operations, namely significance coding operation, sign coding operation, magnitude refinement coding operation, and cleanup coding operation are performed in the corresponding coding passes. For a transform coefficient that is currently insignificant, if any of the 8 neighboring transform coefficients are already significant, it is coded in the significance propagation pass using the significance coding operation; otherwise, it is coded in the cleanup pass using the cleanup coding operation; if this coefficient becomes significant, the sign is coded immediately using the sign coding operation. In the magnitude refinement pass, the

magnitude of the significant transform coefficients that have been found in previous coding passes is updated using the magnitude refinement coding operation. The output bit streams of coding passes can be further coded by using an arithmetic coding engine to improve the compression performances at the cost of computational complexity. Based on the current status of the 8 neighboring transform coefficients, JPEG2000 defines 18 context labels, and stores their respective probability modes in the MQ table. More precisely, 10 context labels are defined for the significance coding operation and the cleanup coding operation, 5 context labels for the sign coding operation, and 3 context labels for the magnitude refinement coding operation.

3 The CBRDE Algorithm

In JPEG2000, a large image is first divided into rectangular sub-images called tiles; each tile is decomposed into subbands by wavelet transform; every subband is partitioned further into small blocks called code blocks, which are quantized to form bit-planes and then coded by EBCOT from the most significant bit-plane to the least significant bit-plane. For each bit-plane, all the code blocks of an image must be processed in the first tier of EBCOT before proceeding with the application of the PCRD algorithm in the second tier. As code blocks of less importance are not needed for the optimal decoded image at a given bit rate, waste of computational power and memory space might result. In this section, an efficient scheme is proposed for the scanning order of code blocks such that waste of computational power and memory space can be reduced.

3.1 Adaptive scanning order

Recall that the code blocks of an image are coded bit-plane by bit-plane, from most to least significant, and the output bitstream can be truncated at an intermediate point between bit-planes; it raises the following interesting questions regarding the scanning order of code blocks. For each bit-plane, is there an adaptive scanning order such that the available coding bits can be allocated for the most significant code block? Is there a common piece of information available at both encoder and decoder, based on which code blocks can be arranged adaptively in decreasing order of significance? If so, there is no need to store all the code streams of code blocks and transmit the scanning order from encoder

to decoder. Is there any simple way to solve the above-mentioned questions?

As one might expect, the first question can be solved effectively by using the rate distortion curve of code blocks. The optimal scanning strategy turns out to take the code block with the steepest rate distortion slope as the first to be coded. In other words, the available coding bits should be first allocated for the code block with the largest amount of distortion decrease per coding bit. Take the second question into consideration; estimated rate distortion slope is preferable to the true one for arranging the scanning order of code blocks. Since the MQ table of probabilities of more probable symbol (MPS) and less probable symbol (LPS) is available at both encoder and decoder, an efficient, context based rate distortion estimation (CBRDE) is therefore proposed to estimate the rate distortion slope of code blocks, which is given in the following subsection.

3.2 Proposed CBRDE

For each bit-plane, code blocks with significant rates of distortion decrease per coding bit should be coded as early as possible. In order to avoid transmitting the scanning order of code blocks from encoder to decoder, the estimated rate distortion slope of a code block is thus adopted, which is given by

$$S = \frac{E[\Delta D]}{E[\Delta R]} \quad (3)$$

where ΔD and ΔR denote the amount of distortion decrease and the number of coding bits, respectively, and $E[\cdot]$ is an expectation operator. Though most of correlation between images pixels can be removed by wavelet transform, there may still be residual correlation between neighboring wavelet coefficients. To incorporate with the framework of EBCOT, the proposed CBRDE for code blocks is based on the MQ table, which is available at both encoder and decoder. Specifically, let B_i denote the i^{th} code block (in wavelet domain), $b_{ij}(m,n)$ the j^{th} significant bit of a coefficient at position: (m,n) in B_i , and $B_{ij} = \bigcup_{m,n} \{b_{ij}(m,n)\}$. The proposed CBRDE for B_{ij} is as follows.

$$S_{ij} = \frac{\sum_{m,n} \text{prob}(b_{ij}(m,n) = 1)}{\sum_{m,n} H(b_{ij}(m,n))} \quad (4)$$

where $H(b_{ij}(m,n))$ denotes the entropy of $b_{ij}(m,n)$ given by

$$H(b) = -\text{prob}(b=1) \cdot \log_2(\text{prob}(b=1)) - \text{prob}(b=0) \cdot \log_2(\text{prob}(b=0)) \quad (5)$$

The numerator of equation (4) is proportional to the amount of distortion decrease; and the denominator is proportional to the number of coding bits. It is noted that probability models of $b_{ij}(m,n)$ can be obtained directly from the MQ table, and therefore the computational complexity is nothing but one lookup table operation.

Take the commonly used Lena image as an example, the true distortion: ΔD and coding bits: ΔR , and their respective estimates $E[\Delta D]$ and $E[\Delta R]$ (for a given bit-plane) are shown in Fig. 2. As one can see, the nearly proportional relationship between the horizontal axis (true values) and the vertical axis (estimated values) demonstrates the potential of the proposed CBRDE.

3.3 Embedded block coding with CBRDE

Figure 3 depicts the proposed image encoder using embedded block coding (EBC) with adaptive block ordering (ABO). In which, the EBC algorithm is the same as the first tier of EBCOT; however, ABO with CBRDE replaces the second tier of EBCOT, i.e. PCR. With the adaptation of the MQ table, the code blocks of an image can be arranged according to their respective estimated rate distortion slopes and then coded in an adaptive manner. Figure 4 depicts the image decoder. Recall that the rate distortion slopes of code blocks can be also estimated at decoder. Thus, the overhead of transmitting the scanning order of code blocks from encoder to decoder is unnecessary; this is beneficial to the communication applications.

4 Experimental Results

The proposed CBRDE for arranging the scanning order of code blocks is valuated on grayscale images. The test images, namely Barbara and Fingerprint are shown in Fig. 5. The biorthogonal Daubechies wavelet with 9/7-tap filters is used. The compression rate is measured in bits per pixel (bpp). The distortion is defined as mean squared error (MSE). The compression rates and MSE values are used to plot the rate distortion curves. To avoid the overhead of transmitting the contributions of code blocks, the second tier of EBCOT, i.e. PCR is replaced by the CBRDE-based ABO. For comparison, as most of images' energies are concentrated in the low frequency subbands, the first tier of EBCOT incorporated with a fixed scanning order starting from the lowest frequency subband is evaluated.

More specifically, the wavelet subbands of an image are coded in a zigzag order from the lowest frequency subband to the highest frequency subband; the code blocks within each subband are scanned from left to right, top to bottom.

For images with textures such as Barbara, there are many significant wavelet coefficients in the middle and high frequency subbands. As shown in Fig. 6, the coding performance can be significantly improved at low bit rates. Figure 7 shows the simulation results of Fingerprint image. It is noted that the rate distortion curve obtained by the proposed image coder is almost convex; this demonstrates the benefit of the proposed CBRDE when the embedded code stream of an image is truncated at an intermediate point between bit-planes.

5 Conclusion

Wavelet transform has been adopted by JPEG2000 as the underlying method to decompose an image into subbands with orientation selectivity. It provides many desirable properties, e.g. multiresolution analysis, high correlation across wavelet subbands of the same orientation, and energy clustering within each subband; these properties are suitable for the image compression applications. In EBCOT, all the code blocks of an image are coded to generate a set of code streams with their respective contributions to the decoded image, based on which the optimal code stream can be obtained by concatenating the suitably truncated code streams through the PCRD algorithm. As some code blocks, which are less important, are not needed for the optimal decoded image at a given bit rate, waste of computational power and memory space may result. Furthermore, implementation of the PCRD algorithm is one of the crucial issues. To overcome the above-mentioned problems, a simple context-based rate distortion estimation (CBRDE) is proposed to arrange the scanning order of code blocks in an adaptive manner. To avoid transmitting the side information regarding the scanning order of code blocks from encoder to decoder, the proposed CBRDE is based on the MQ table, which is available at both encoder and decoder. The second tier of EBCOT, i.e. PCRD can therefore be replaced by the CBRDE-based adaptive block ordering (ABO). Experimental results show that the rate distortion curves are almost convex.

Recall that for each code block, the coding procedure proceeds in distinct passes. Thus, the proposed image coding with adaptive block ordering can be extended to image coding with adaptive pass

ordering, however, at the cost of increasing complexity from the implementation point of view.

References:

- [1] J. M. Shapiro, "Embedded Image Coding Using Zero-Trees of Wavelet Coefficients," *IEEE Trans. On Signal Processing*, vol. 40, pp. 3445-3462, 1993.
- [2] A. Said and W. A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. On Circuits Syst. Video Tech.* vol. 6, pp. 243-250, 1996.
- [3] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, Low Complexity Image Coding With a Set-Partitioning Embedded Block Coder," *IEEE Trans. On Circuits Syst. Video Tech.* vol. 14, pp. 1219-1235, Nov., 2004.
- [4] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Trans. On Image Processing*, vol. 9, pp. 1158-1170, July, 2000.
- [5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, pp. 36-58, September, 2001.
- [6] H.-C. Fang, Y.-W. Chang, T.-C. Wang, C.-T. Huang, and L.-G. Chen, "High-Performance JPEG 2000 Encoder with Rate-Distortion Optimization," *IEEE Trans. On Multimedia*, vol. 8, no. 4, pp. 645-653, August. 2006.

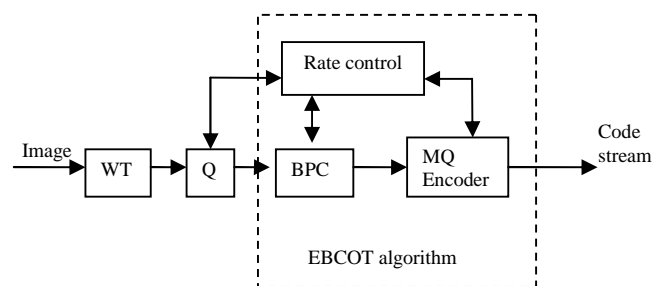
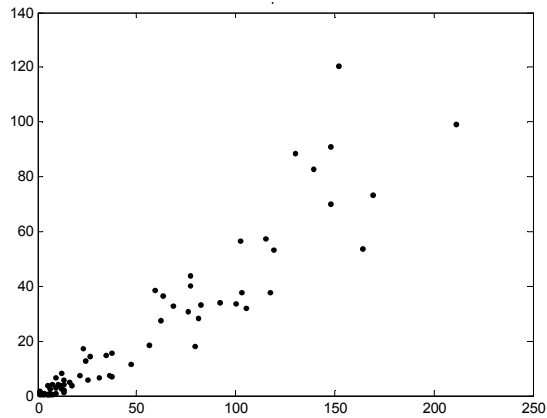
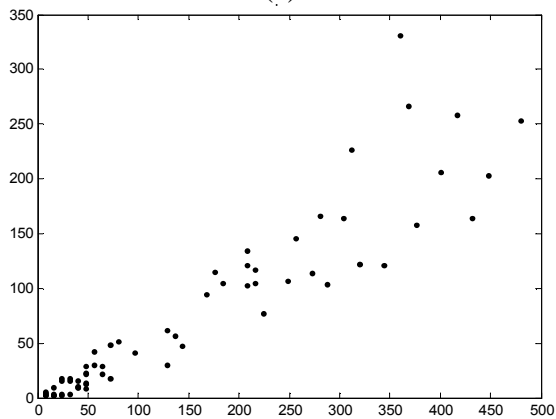


Fig. 1 Block diagram of JPEG2000 Encoder



(a)



(b)

Fig. 2 Performance of CBRDE applied to Lena image; (a) horizontal axis: true ΔD , vertical axis: estimated $E[\Delta D]$; (b) horizontal axis: true ΔR , vertical axis: estimated $E[\Delta R]$.

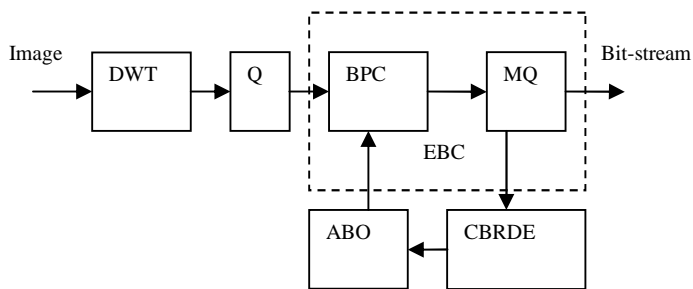


Fig. 3 Proposed image encoder using EBC with ABO

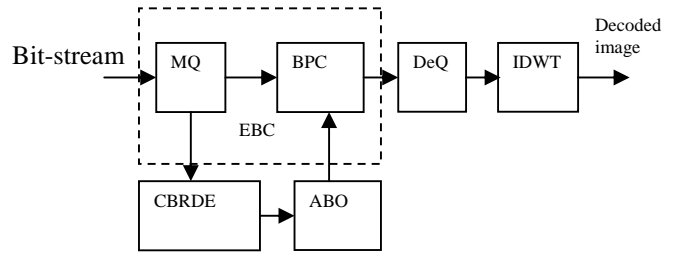


Fig. 4 Proposed image decoder using EBC with ABO



(a)



(b)

Fig. 5 Test images (a) Barbara (b) Fingerprint

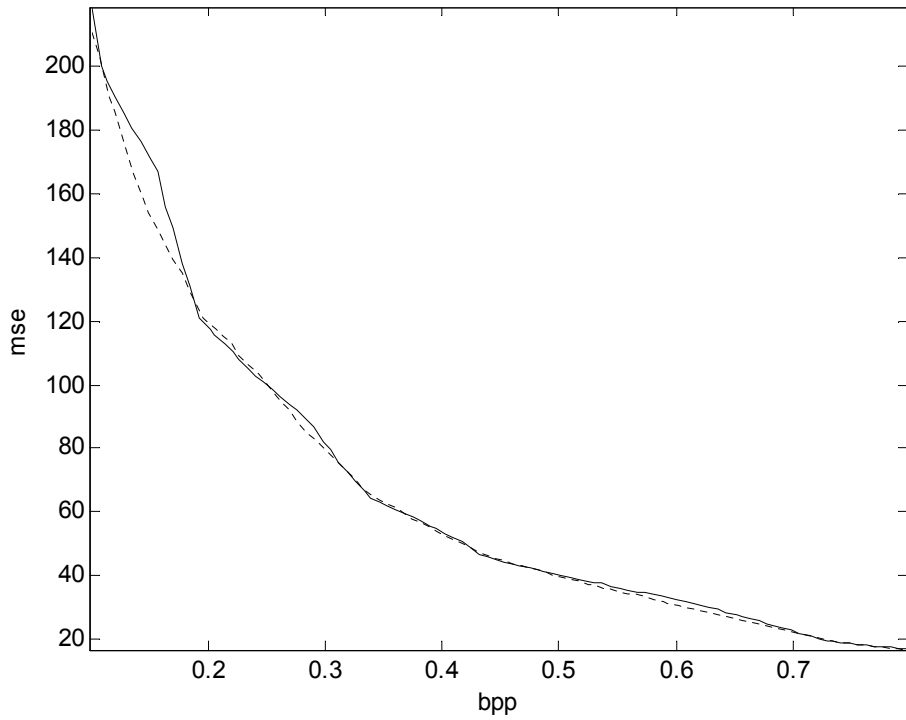


Fig. 6 Rate distortion curves of Barbara image by EBC with the CBRDE-based ABO (dashed line) and EBC with a fixed scan order (solid line); Vertical axis: mean square error (MSE); Horizontal axis: bit rate (bpp).

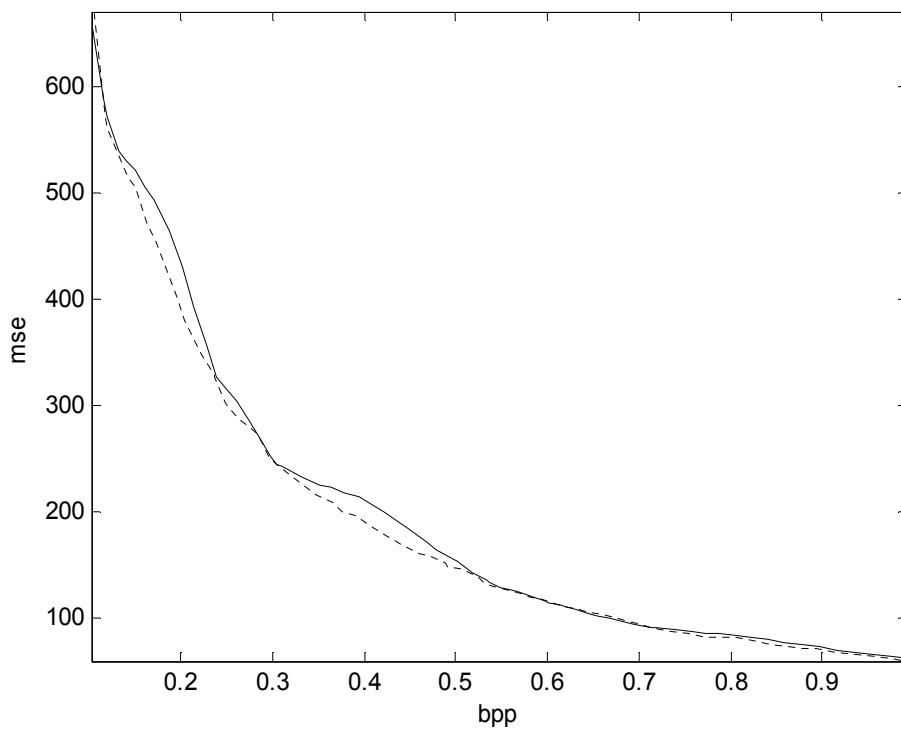


Fig. 7 Rate distortion curves of Fingerprint image by EBC with the CBRDE-based ABO (dashed line) and EBC with a fixed scan order (solid line); Vertical axis: mean square error (MSE); Horizontal axis: bit rate (bpp).

Reconfigurable VLSI Architecture for FFT Processor

Tze-Yun Sung
 Department of Microelectronics
 Engineering
 Chung Hua University
 707, Sec. 2, Wufu Road
 Hsinchu City 300-12, Taiwan
 bobsung@chu.edu.tw

Hsi-Chin Hsin
 Department of Computer Science
 and Information Engineering
 National United University
 Miaoli 36003, Taiwan
 hsin@nuu.edu.tw

Abstract: - This paper presents a CORDIC (Coordinate Rotation Digital Computer)-based split-radix fast Fourier transform (FFT) core for OFDM systems, for example, Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL), and Worldwide Interoperability for Microwave Access (WiMAX). High-speed 128/256/512/1024/ 2048/4096/8192-point FFT processors have been implemented by 0.18 (1p6m) at 1.8V, in which all the control signals are generated internally. These FFT processors outperform the conventional ones in terms of both power consumption and core area.

Key-Words: - IP, FFT, split-radix, CORDIC, OFDM.

1 Introduction

High-performance fast Fourier transform (FFT) processor is needed especially for real-time digital signal processing applications. Specifically, the computation of discrete Fourier transform (DFT) ranging from 128 to 8192 points is required for the orthogonal frequency division multiplexer (OFDM) of the following standards: Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL) and Worldwide Interoperability for Microwave Access (WiMAX) [1]-[8]. Thompson [9] proposed an efficient VLSI architecture for FFT in 1983. Wold and Despain [10] proposed pipelined and parallel-pipelined FFT for VLSI implementations in 1984. Widhe [11] developed efficient processing elements of FFT in 1997. To reduce the computation complexity, the split-radix 2/4, 2/8, and 2/16 FFT algorithms were proposed in [12]-[15].

As the Booth multiplier is not suitable for hardware implementations of large FFT, we propose the CORDIC-based multiplier. Moreover, we develop a ROM-free twiddle factor generator using simple shifters and adders only [1], which obviates the need to store all the twiddle factors in a large ROM space. As a result, the proposed CORDIC-based split-radix FFT core with the ROM-free twiddle factor generator is suitable for the wireless local area network

(WLAN) applications. In this paper, a high-performance 128/256/512/1024/2048/4096/8192-point FFT processor is presented for the European and Japanese standards. The remainder of this paper proceeds as follows. In Section II, the split-radix 2/8 FFT algorithm and the CORDIC algorithm are reviewed briefly. In Section III, the reusable IP 128-point CORDIC-based split-radix FFT core is proposed. In Section IV, the hardware implementations of FFT processors are described. The performance analysis is presented in Section V. Finally, the conclusion is given in Section VI.

2 Review of Split-Radix FFT and CORDIC Algorithm

2.1 Split-Radix FFT

The idea behind the split-radix FFT algorithm is to compute the even and odd terms of FFT separately. The even term of the split-radix 2/8 FFT algorithm is given by

$$X(2k) = \sum_{n=0}^{N/2-1} (x(n) + x(n + \frac{N}{2}))W_{N/2}^{nk} \quad (1)$$

where $W_{N/2} = e^{-j\frac{2\pi}{N/2}}$ and $k = 0, 1, 2, \dots, (N/2) - 1$.

The odd term is as follows:

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\begin{aligned}
 X(8k+l) = & \sum_{n=0}^{N/8-1} \left((x(n) + x(n + \frac{2N}{8})W_4^l + x(n + \frac{4N}{8})W_4^{2l} \right. \\
 & + x(n + \frac{6N}{8})W_4^{3l} + x(n + \frac{N}{8}) + x(n + \frac{3N}{8})W_4^l \quad (2) \\
 & \left. + x(n + \frac{5N}{8})W_4^{2l} + x(n + \frac{7N}{8})W_4^{3l} \right) W_N^{nl} W_{N/8}^{nk}
 \end{aligned}$$

where $k = 0, 1, 2, \dots, (N/8) - 1$ and $l = 1, 3, 5, 7$. The split-radix 2/8 FFT algorithm, which combined with radix-2 and radix-4 proves effective to develop a reusable IP 128-point FFT core.

2.2 CORDIC Algorithm

The CORDIC algorithm in the circular coordinate system is as follows [16].

$$x(i+1) = x(i) - \sigma_i 2^{-i} y(i) \quad (3)$$

$$y(i+1) = y(i) + \sigma_i 2^{-j} x(i) \quad (4)$$

$$z(i+1) = z(i) - \sigma_i \alpha(i) \quad (5)$$

$$\alpha(i) = \tan^{-1} 2^{-i} \quad (6)$$

where $\sigma_i = \text{sign}(z(i))$ with $z(i) \rightarrow 0$ in the rotation mode, and $\sigma_i = -\text{sign}(x(i)) \cdot \text{sign}(y(i))$ with $y(i) \rightarrow 0$ in the vectoring mode. The scale factor:

$k(i)$ is equal to $\sqrt{1 + \sigma_i^2 2^{-2i}}$. After n micro-rotations, the product of the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k(i) = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \quad (7)$$

Notice that CORDIC in the circular coordinate system with rotation mode can be written by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = K_c \begin{bmatrix} \cos z_0 & \sin z_0 \\ -\sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (8)$$

where $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ and $\begin{bmatrix} x_n \\ y_n \end{bmatrix}$ are the input vector and the

output vector, respectively, z_0 is the rotation angle, and K_c is the scale factor. In [1], the circular rotation computation of CORDIC was used for complex multiplication with $e^{-j\theta}$, which is given by

$$\begin{bmatrix} \text{Re}[X'] \\ \text{Im}[X'] \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \text{Re}[X] \\ \text{Im}[X] \end{bmatrix} \quad (9)$$

3 Reusable IP 128-Point CORDIC-Based Split-Radix FFT Core

Figure 1 shows the proposed 128-point CORDIC-based split-radix FFT processor, which can be used as a reusable IP core for various FFT with multiples of 128 points. Notice that the modified split-radix 2/8 FFT butterfly processor and the

ROM-free twiddle factor generator are used. In addition, an internal (128 32-bit) SRAM is used to store the input and output data for hardware efficiency, through the use of the in-place computation algorithm [1]

3.1 CORDIC-Based Split-Radix 2/8 FFT Processor

For the butterfly computation of the proposed CORDIC-based split-radix 2/8 FFT processor, sixteen complex additions, two constant multiplications (CM), and four CORDIC operations are needed, as shown in Figure 2. The CORDIC algorithm has been widely used in various DSP applications because of the hardware simplicity. According to equation (9), the twiddle factor multiplication of FFT can be considered a 2-D vector rotation in the circular coordinate system. Thus, CORDIC in the circular coordinate system with rotation mode is adopted to compute complex multiplications of FFT.

The pipelined CORDIC arithmetic unit can be obtained by decomposing the CORDIC algorithm into a sequence of operational stages. In [17], we derived the error analysis of fixed-point CORDIC arithmetic, based on which, the number of the CORDIC stages can be determined effectively. For example, the number of the CORDIC stages is 12 if the overall relative error of 16-bit CORDIC arithmetic is required to be less than 10^{-3} . The pipelined CORDIC arithmetic unit with 12 stages and an additional pre-scalar stage. In which, the pre-calculated scaling factor $K_c \approx 1.64676$ and the Booth binary recoded format leads to 1.101001. The main concern for the design of the CORDIC arithmetic unit is throughput rather than latency. The proposed CORDIC arithmetic unit in terms of gate counts is less than 4 real multipliers significantly. In addition, the power consumption can be reduced significantly by using the proposed CORDIC arithmetic unit; it has been reduced by 30% according to the report of PrimePower® distributed by Synopsys.

As the twiddle factors: W_8^1 and W_8^3 are equal to $\frac{\sqrt{2}}{2}(1-j)$ and $-\frac{\sqrt{2}}{2}(1+j)$, respectively, a complex number, say $(a+bj)$, times W_8^1 or W_8^3 can be written by

$$(a+bj) \times \left(\frac{\sqrt{2}}{2}(1-j) \right) = \frac{\sqrt{2}}{2} ((a+b) + j(-a+b)) \quad (10)$$

$$(a+bj) \times \left(-\frac{\sqrt{2}}{2}(1+j) \right) = -\frac{\sqrt{2}}{2} ((a-b) + j(a+b)) \quad (11)$$

where $\frac{\sqrt{2}}{2}$ can be represented as $1.0\bar{1}0\bar{1}010$ using the Booth binary recoded form (BBRF). Thus, the CM unit can be implemented by using simple adders and shifters only. Figure 3 shows the pipelined CM architecture, which uses three subtractions/additions and therefore improves on the computation speed significantly.

Based on the above-mentioned CORDIC arithmetic unit and CM unit, the computational circuit and hardware architecture of the CORDIC-based split-radix 2/8 FFT butterfly computation are realized. As one can see, the pipelined CORDIC arithmetic unit aims at increasing the throughput of complex multiplications..

3.2 ROM-Free Twiddle Factor Generator

In the conventional FFT processor, a large ROM space is needed to store all the twiddle factors. To reduce the chip area, a twiddle factor generator is thus proposed. Figure 4 shows the ROM-free twiddle factor generator using simple adders and shifters for 128-point FFT. In which, the 16-bit accumulator is to generate the value $2n\pi$ for each index n ; $n = 2^{\log_2 N - 3} - 1$, the 16-bit shifter is to divide $2n\pi$ by N , and the 16-bit shifter/adder is to produce the twiddle factors: θ_N^{1n} , θ_N^{3n} , θ_N^{5n} and θ_N^{7n} . By using the twiddle factor generator, the chip area and power consumption can be reduced significantly at the cost of an additional logic circuit. Table 1 shows the gate counts of the full-ROM storing all the twiddle factors, the CORDIC twiddle factor generator [1] and the ROM-free twiddle factor generator.

4 Implementation of FFT Processors

The 128/256/512/1024/2048/4096/8192-point FFT processors. In which, the radix-2 and split-radix 2/4 butterfly processors [1] using the pipelined CORDIC arithmetic units and twiddle factor generators are implemented; and moreover, two memory banks (4096/2048/1024/512/256/0 \times 32-bit and 8192/4096/2048/1024/512/256/128 \times 32-bit) are allocated for increased efficiency by using the in-place computation algorithm [1]. Hardware architecture is shown in Figure 5.

The hardware code written in Verilog[®] is running on a workstation with the ModelSim[®] simulation tool and Synopsys[®] synthesis tool (design compiler). The chips are synthesized by the TSMC 0.18 μm 1p6m CMOS cell libraries [18]. The physical circuit is synthesized by the Astro[®] tool. The circuits are evaluated by DRC, LVS and PVS [19].

The layout view of the 8192-point FFT processor is

shown in Figure 6. The core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point and 8192-point FFT processors are shown in Table 2. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

5 Performance Analysis of The Proposed FFT Architecture

FFT processors used to compute 128/256/512/1024/2048/4096/8192-point FFT are composed mainly of the 128-point CORDIC-based split-radix 2/8 FFT core; the computation complexity using a single 128-point FFT core is $O(N/6)$ for N -point FFT. The log-log plot of the CORDIC computations versus the number of FFT points is shown in Figure 7. As one can see, the proposed FFT architecture is able to improve the power consumption and computation speed significantly.

6 Conclusion

This paper presents low-power and high-speed FFT processors based on CORDIC and split-radix techniques for OFDM systems. The architectures are mainly based on a reusable IP 128-point CORDIC-based split-radix FFT core. The pipelined CORDIC arithmetic unit is used to compute the complex multiplications involved in FFT, and moreover the required twiddle factors are obtained by using the proposed ROM-free twiddle factor generator rather than storing them in a large ROM space.

The CORDIC-based 128/256/512/1024/2048/4096/8192-point FFT processors have been implemented by 0.18 μm CMOS, which take 395 μs , 176.8 μs , 77.9 μs , 33.6 μs , 14 μs , 5.5 μs and 1.88 μs to compute 8192-point, 4096-point, 2048-point, 1024-point, 512-point, 256-point and 128-point FFT, respectively.

The CORDIC-based FFT processors are designed by using the portable and reusable Verilog[®]. The 128-point FFT core is a reusable intellectual property (IP), which can be implemented in various processes and combined with an efficient use of hardware resources for the trade-offs of performance, area, and power consumption.

References:

- [1] T. Y. Sung, "Memory-efficient and high-speed split-radix FFT/IFFT processor based on

- pipelined CORDIC rotations,” *IEE Proc.-Vis. Image Signal Proc.*, Vol. 153, No. 4, Aug. 2006, pp.405-410.
- [2] J. C. Kuo, C. H. Wen, A. Y. Wu, “Implementation of a programmable 64/spl sim/2048-point FFT/IFFT processor for OFDM-based communication systems,” *Proceedings of the 2003 International Symposium on Circuits and Systems*, Volume 2, 25-28 May 2003 pp.II-121 - II-124.
- [3] L. Xiaojin, Z. Lai, C. J. Cui, “A low power and small area FFT processor for OFDM demodulator,” *IEEE Transactions on Consumer Electronics*, Volume 53, Issue 2, May 2007, pp. 274 – 277.
- [4] J. Lee, H. Lee, S. I. Cho, S. S. Choi, “A high-speed, low-complexity radix-216 FFT processor for MB-OFDM UWB systems,” *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp.
- [5] A. Cortes, I. Velez, J. F. Sevillano, A. Irizar, “An approach to simplify the design of IFFT/FFT cores for OFDM systems,” *IEEE Transactions on Consumer Electronics*, Volume 52, Issue 1, Feb. 2006, pp.26 – 32.
- [6] Y. H. Lee, T. H. Yu, K. K. Huang, A. Y. Wu, “Rapid IP design of variable-length cached-FFT processor for OFDM-based communication systems,” *IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 2006 pp.62-65.
- [7] C. L. Wey, W. C. Tang, S. Y. Lin, “Efficient memory-based FFT architectures for digital video broadcasting (DVB-T/H),” *2007 International Symposium on VLSI Design, Automation and Test*, 25-27 April 2007, pp.1-4.
- [8] Y. W. Lin, H. Y. Liu, C. Y. Lee, “A 1-GS/s FFT/IFFT processor for UWB applications,” *IEEE Journal of Solid-State Circuits*, Volume 40, Issue 8, Aug. 2005, pp.1726-1735.
- [9] C. D. Thompson, “Fourier transform in VLSI,” *IEEE Transactions on Computers*, Vol.32, No. 11, 1983, pp.1047-1057.
- [10] E. H. Wold, A. M. Despain, “Pipelined and parallel-pipelined FFT processor for VLSI implementation,” *IEEE Transactions on Computers*, Vol.33, No. 5, 1984, pp.414-426.
- [11] T. Widhe, “Efficient implementation of FFT processing elements,” *Linkoping Studies in Science and Technology*, Thesis No. 619, Linkoping University, Sweden, 1997.
- [12] P. Duhamel, H. Hollmann, “Implementation of "split-radix" FFT algorithms for complex, real, and real symmetric data.” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Volume 10, April 1985, pp.784 – 787.
- [13] A .A. Petrovsky, S. L. Shkredov, “Automatic generation of split-radix 2-4 parallel-pipeline FFT processors: hardware reconfiguration and core optimizations,” *2006 International Symposium on Parallel Computing in Electrical Engineering*, pp.181-186.
- [14] S. Bouguezal, M. O. Ahmad, M. N. S. Swamy, “A new radix-2/8 FFT algorithm for length-q/spl times/2/sup m/ DFTs,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume 51, Issue 9, 2004, pp.1723- 1732.
- [15] W. C. Yeh, C. W. Jen, “High-speed and low-power split-radix FFT.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 51, Issue 3, March 2003, pp.864 – 874.
- [16] M. D. Ercegovac, T. Lang, “CORDIC algorithm and implementations.” Digital Arithmetic, Morgan Kaufmann Publishers, 2004, Chapter 11.
- [17] T. Y. Sung, H. C. Hsin, “Fixed-point error analysis of CORDIC arithmetic for special-purpose signal processors,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E90-A, No.9, Sep. 2007, pp.2006-2013.
- [18] “TSMC 0.18 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.
- [19] Cadence design systems: [http://www.cadence.com/products /pages/default.aspx](http://www.cadence.com/products/pages/default.aspx).

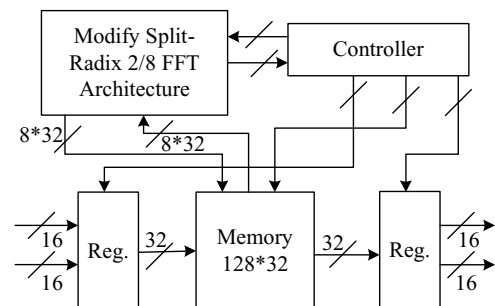


Fig. 1 The proposed 128-point CORDIC- based split-radix FFT processor

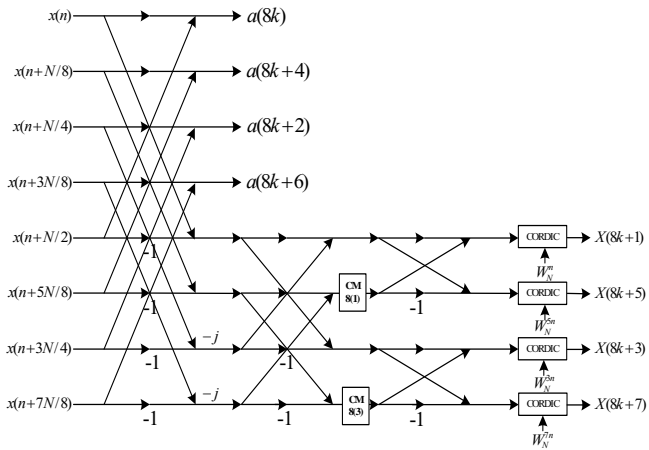


Fig. 2 Data flow of the butterfly computation of the modified split-radix 2/8 FFT

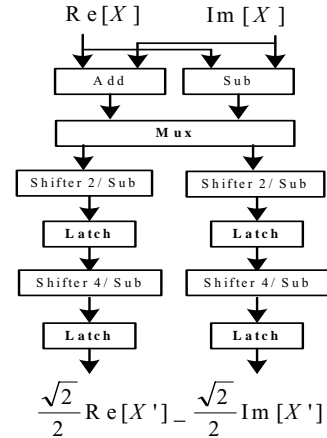


Fig. 3 Constant multiplier (CM) architecture for the modified split-radix 2/8 FFT

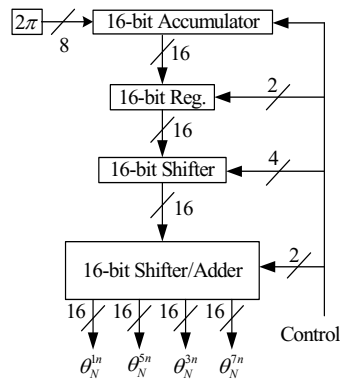


Fig. 4 Proposed ROM-free twiddle factor generator for 128-point FFT

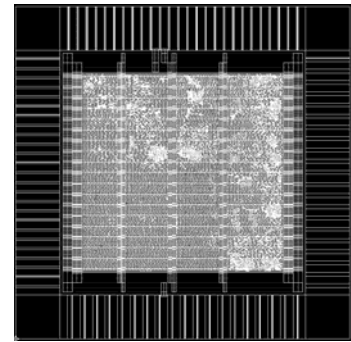


Fig. 6 Layout view of the 8192-point FFT processor

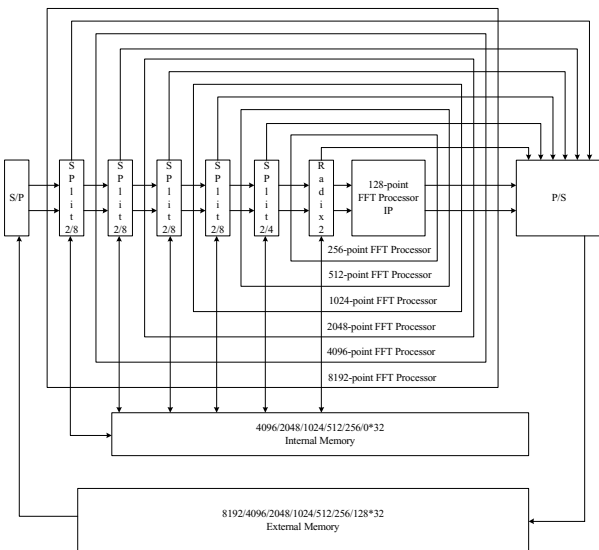


Fig. 5 Hardware architecture of the 128/256/512/1024/2048/4096/8192-point FFT processor

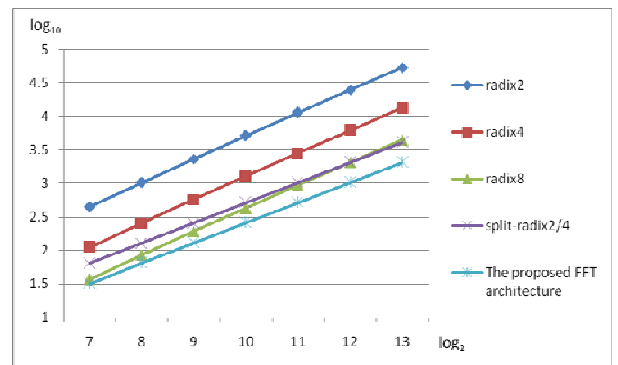


Fig. 7 Log-log plot of the CORDIC computations versus the number of FFT points

Table 1 Hardware requirements of the full-ROM, the CORDIC twiddle factor generator [1], and the ROM-free twiddle factor generator

Full-Twiddle Factor ROM					1bit~1gate
8192-Point ROM					
4K×16 bit					
CORDIC Twiddle Factor Generator (T. Y. Sung, 2006) [1]					
16-bit CORDIC ~18K bit	11-bit Adder ~150 gates	11-bit Shifter ~50 gates	16-bit Shifter ~90 gates	16-bit Adder ~200 gates	
ROM-free Twiddle Factor Generator (Sung, Hsin and Cheng, 2008)					
16-bit Accumulator ~200gates	16-bit Register ~32 gates	16-bit Shifter ~90 gates	16-bit Shifter/Adder ~90×2+200×2 gates		

Table 2 Core areas, power consumptions, clock rates of 128-, 256-, 512-, 1024-, 2048-, 4096- and 8192-point FFT generator

FFT Size	Core Area	Power Consumption	Clock Rate
128-point	2.28mm ²	80mW	200MHz
256-point	2.37mm ²	84mW	200MHz
512-point	2.49mm ²	88mW	200MHz
1024-point	2.62mm ²	94mW	200MHz
2048-point	2.81mm ²	99mW	200MHz
4096-point	3.10mm ²	106mW	200MHz
8192-point	3.62mm ²	117mW	200MHz

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

98 年 05 月 26 日

附件三

報告人姓名	宋志雲	服務機構 及職稱	中華大學微電子工程學系 副教授
時間 會議 地點	98.05.20~98.05.22 中國 杭州市	本會核定 補助文號	計畫編號: NSC 97-2221-E-216-044
會議 名稱	(英文) 9 th WSEAS International Conference on Multimedia Systems & Signal Processing 8 th WSEAS International Conference on Instrumentation, Measurement, Circuit and Systems		
發表 論文 題目	(英文) High-SFDR and Multiplierless Direct Digital Frequency Synthesizer, A Simple Rate Distortion Estimation for Embedded Block Coding, Reconfigurable VLSI Architecture for FFT processor.		

報告內容應包括下列各項：

一、參加會議經過

發表論文三篇，分別發表於 9th WSEAS International Conference on Multimedia Systems & Signal Processing, 8th WSEAS International Conference on Instrumentation, Measurement, Circuit and Systems 並與與會人士討論及聽取議程主席之意見。本研討會為 WSEAS 之重要研討會，歷史悠久，至少有二十年以上之歷史。本人發表之論文接獲選為最佳論文，並邀請增加篇幅在其期刊(EI)登出。本人發表之論文，引起大家極大興趣，會中廣泛討論內容。

二、與會心得

此一研討會為 WSEAS 重要研討會，與會人士均為一時之選。會中可認識相關領域的領導人物，機會難得。論文如獲最佳論文，延伸後將獲得該學會之期刊(EI)刊登。

三、考察參觀活動(無是項活動者省略)

無

四、建議

鼓勵在教學之學校教師投稿此一研討會，因為有指標性意義。同時爭取 WSEAS 類似權威性研討會之主辦權，以增加台灣之國際學術地位。

五、攜回資料名稱及內容

研討會之論文光碟及紙本論文集。

六、其他

High-SFDR and Multiplierless Direct Digital Frequency Synthesizer

Tze-Yun Sung

Department of
Microelectronics Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
bobsung@chu.edu.tw

Hsi-Chin Hsin

Department of Computer
Science and Information
Engineering
National United University
Miaoli 36003, Taiwan
hsin@nuu.edu.tw

Lyu-Ting Ko

Department of Electrical
Engineering
Chung Hua University
Hsinchu City 300-12, Tawan,
m09601049@chu.edu.tw

Abstract: - This paper presents a hybrid CORDIC (COordinate Rotation DIgital Computer) algorithm for designs and implementations of the direct digital frequency synthesizer (DDFS). The proposed multiplier-less architecture with small ROM (4×16 -bit) and pipelined data path provides a spurious free dynamic range (SFDR) of more than 84.4 dBc. A SoC (system on chip) has been designed by 1P6M CMOS, and then emulated on the Xilinx FPGA. It is shown that the hybrid CORDIC-based architecture is suitable for VLSI implementations of the DDFS in terms of hardware cost, power consumption, and SFDR.

Key-Words: - DDFS, hybrid CORDIC, SoC, FPGA, SFDR.

1 Introduction

The direct digital frequency synthesizer (DDFS) plays a key role in many digital communication systems. Fig. 1 depicts the conventional DDFS, which consists mainly of phase accumulator, sine/cosine generator, digital-to-analog converter, and low-pass filter. The sine/cosine generator as the core of DDFS is usually implemented by using a ROM lookup table; with high spurious free dynamic ranges (SFDR) comes a large ROM lookup table [1]. In order to reduce the size of the lookup table, many techniques were proposed [1]-[4]. The quadrant compression technique can reduce the ROM size by 75% [2]. The Sunderland architecture is to split the ROM into two smaller ones [3], and its improved version known as the Nicholas architecture results in a higher ROM-compression ratio (32:1) [4]. In [5], the polynomial hyperfolding technique with high order polynomial approximation was used to design DDFS. In [6] and [7], the angle rotation algorithm was used to design quadrature direct digital frequency synthesizer/complex mixer (QDDSM). COordinate Rotation DIgital Computer (CORDIC) is a well known arithmetic algorithm, which evaluates various elementary functions including sine and cosine functions by using simple adders and shifters only. Thus, CORDIC is suitable for the design of high-performance chips with VLSI technologies. Recently, the CORDIC algorithm has received a lot of attention to the design of high-performance DDFS [8]-[11], especially for the modern digital

communication systems.

This paper is organized as follows. In section II, the hybrid CORDIC algorithm is proposed. In section III, hardware implementation of DDFS is described. The performance analysis is presented in section IV. Finally, the conclusion is given in section V.

2 The Hybrid CORDIC Algorithm

In this section, the hybrid CORDIC algorithm is proposed, and based on which, a low-power and high-SFDR DDFS can be developed.

2.1 Modified Angle Recoding Method for CORDIC Algorithm

In order to reduce the number of CORDIC iterations, the input angle can be divided into encoded angles by using the modified Booth encoding (MBE) method [12]. Specifically, let ψ denote the input angle represented by

$$\psi = f(0)2^{-p} + f(1)2^{-(p+1)} + \dots + f(w-1)2^{-(w-1)} \quad (1)$$

where $f(i) \in \{0,1\}$, w is the word length of operands, and $\left\lceil \frac{(w-2.585)}{3} \right\rceil = p \leq i \leq w-1$. The MBE decomposition of ψ is as follows.

$$\psi = \sum_{i=p/2}^{(w-1)/2} \beta(i) \quad (2)$$

where the encoded angle:

$$\beta(i) = \rho(i)2^{-i} \text{ with } \rho(i) \in \{-2,-1,0,1,2\}. \text{ As } \sin \beta(i)$$

and $\cos \beta(i)$ can be approximated by

$$\sin \beta(i) \cong \beta(i) = \rho(i)2^{-i} \quad (3)$$

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\cos \beta(i) \cong 1 - \frac{\beta^2(i)}{2} = 1 - \rho^2(i)2^{-(2i+1)} \quad (4)$$

we have

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 - \rho^2(i)2^{-(2i+1)} & \rho(i)2^{-i} \\ -\rho(i)2^{-i} & 1 - \rho^2(i)2^{-(2i+1)} \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (5)$$

$$z(i+1) = z(i) - \rho(i)2^{-i} \quad (6)$$

Fig. 2 shows the proposed architecture for the modified scaling-free CORDIC arithmetic, in which, eight shifters, two CSAs, two CLAs, two latches, and four MUXs are used; the shifters and MUXs are to determine $\rho(i)$.

2.2 The modified scaling-free radix-8 CORDIC Algorithm

By using the modified angle recoding method [12]-[13], the input angle ψ can be divided as follows.

$$\psi = \sum_{i=p}^{w-1} \phi(i) \tan^{-1} 2^{-i} \quad (7)$$

where $\phi(i) \in \{0,1\}$, and w is the word length. The CORDIC iteration is therefore represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} 1 & \phi(i)2^{-i} \\ -\phi(i)2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (8)$$

$$z(i+1) = z(i) - \phi(i) \tan^{-1} 2^{-i} \quad (9)$$

Let $i = 3n - c$; $c \in \{0,1,2\}$. By using the Taylor series expansion, the absolute difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$ is given by

$$\zeta = \left| \tan^{-1} 2^{-(3n-c)} - 2^c \tan^{-1} 2^{-3n} \right| = \left| \frac{1}{3} \cdot 2^{-3(3n-c)} + \Lambda \right| \quad (10)$$

where Λ is the remaining terms of the difference between $\tan^{-1}(2^{-(3n-c)})$ and $2^c \tan^{-1}(2^{-3n})$. Thus, we have

$$\zeta \cong \frac{2^{-3(3n-c)}}{3} = \frac{2^{-3i}}{3} \quad (11)$$

For w -bit operands, ζ can be ignored in the following sense

$$\zeta \leq 2^{-w} \quad (12)$$

Based on equations (11) and (12), we have

$$\frac{2^{-3i}}{3} \leq 2^{-w} \quad (13)$$

$$3i + \log_2 3 \geq w \quad (14)$$

$$i \geq \frac{w - \log_2 3}{3} = \left\lceil \frac{w - \log_2 3}{3} \right\rceil \cong \frac{w}{3} \quad (15)$$

As a result, when $i > \frac{w}{3}$, three consecutive terms of equation (7) can be integrated into a single term as follows:

$$\begin{aligned} & \phi(3n-2) \tan^{-1}(2^{-(3n-2)}) + \phi(3n-1) \tan^{-1}(2^{-(3n-1)}) \\ & + \phi(3n) \tan^{-1}(2^{-(3n)}) \\ & = (\phi(3n-2) \cdot 2^2 + \phi(3n-1) \cdot 2^1 + \phi(3n) \cdot 2^0) \tan^{-1}(2^{-3n}) \\ & = \varphi(n) \tan^{-1}(2^{-3n}) \end{aligned} \quad (16)$$

where $\phi(\cdot) \in \{0,1\}$, and therefore $\varphi(n) \in \{0,1,2,\dots,7\}$. It follows that the resulting radix-8 CORDIC algorithm is represented as

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = K_8(i) \begin{bmatrix} 1 & -\varphi(i) \cdot 2^{-3i} \\ \varphi(i) \cdot 2^{-3i} & 1 \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (17)$$

$$z(i+1) = z(i) - \varphi(i) \tan^{-1} 2^{-3i} \quad (18)$$

$$K_8(i) = (1 + \varphi^2(i) 2^{-6i})^{1/2} \quad (19)$$

The scaling factor K_8 is given by

$$K_8 = \prod_{i=p}^{i=\lceil w/3 \rceil - 1} K_8(i) \quad (20)$$

It can be shown that the scaling factor turns out to be equal to 1 when the input angle is less than $2^{-w/2}$, and moreover, if the input angle is less than $2^{-w/3}$, equation (18) can be rewritten as [19]

$$z(i+1) = z(i) - \varphi(i) 2^{-3i} \quad (21)$$

Fig. 3 depicts the proposed architecture for the modified scaling-free radix-8 CORDIC arithmetic. In which, six shifters, two CSAs, two CLAs, and two latches are used; the shifters and switches are to determine the radices for computations. Note that the number of processors is reduced, and system throughput is increased at the cost of hardware complexity.

2.3 The proposed hybrid CORDIC Algorithm

The input angle Ω can be decomposed into a higher-angle Ω_H and a lower-angle Ω_L represented as

$$\Omega = \Omega_H + \Omega_L = \sum_{i=0}^{\lceil u/2 \rceil - 1} \rho(i) 2^{-2i} + \sum_{i=\lceil u/2 \rceil}^{\lceil u/2 \rceil + \lceil (w-u)/3 \rceil - 1} \varphi(i) 2^{u-1+3(i-\lceil w/4 \rceil)} \quad (22)$$

where w is the word length with the first u bits being the most significant bits; Ω_H and Ω_L are computed by using the modified scaling-free CORDIC algorithm and the modified scaling-free radix-8 CORDIC algorithm, respectively. For computation efficiency, the determination of u is as follows: 1) u must be an odd number to satisfy the MBE method, and 2) $u \geq \frac{w}{2}$ under the scaling-free constraint. Thus, we have

$$u = \begin{cases} 2n+1, & \text{if } w = 4n+0 \\ 2n+1, & \text{if } w = 4n+1 \\ 2n+1, & \text{if } w = 4n+2 \\ 2n+3, & \text{if } w = 4n+3 \end{cases} \quad (23)$$

Based on the above equation, the minimum iteration number of the proposed hybrid CORDIC algorithm can be obtained as shown in Fig. 4. The computations of $x(i)$ and $y(i)$ are therefore as follows.

$$\text{For } \frac{p}{2} \leq i < \left\lceil \frac{u}{2} \right\rceil,$$

$$x(i+1) = x(i) - \rho^2(i)2^{-(4i+1)}x(i) - \rho(i)2^{-2i}y(i) \quad (24)$$

$$y(i+1) = y(i) - \rho^2(i)2^{-(4i+1)}x(i) + \rho(i)2^{-2i}y(i) \quad (25)$$

$$\text{For } \left\lceil \frac{u}{2} \right\rceil \leq i < \left\lceil \frac{w}{4} \right\rceil + \left\lceil \frac{w-u}{3} \right\rceil,$$

$$x(i+1) = x(i) - \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}y(i) \quad (26)$$

$$y(i+1) = y(i) + \varphi(i)2^{u-1+3(i-\lceil w/4 \rceil+1)}x(i) \quad (27)$$

3 Hardware Implementation of the Proposed DDFS

In this section, the DDFS implemented by using the hybrid CORDIC algorithm is presented. Fig. 5 shows the 16-bit DDFS architecture consisting mainly of phase accumulator, phase calculator, and sine/cosine generator, which is different from the conventional architecture. It is noted that the accumulated error in the sine/cosine generator is to be corrected by using the 4×16 -bit correction table. Take into account DAC technology, hardware cost and practical applications, the word length of the propose DDFS is set to 16-bit.

The hybrid CORDIC-based sine/cosine generator with recursively accumulated angle \mathcal{G}_m is given by

$$\begin{bmatrix} x(i+1) \\ y(i+1) \end{bmatrix} = \begin{bmatrix} \cos \mathcal{G}_m & -\sin \mathcal{G}_m \\ \sin \mathcal{G}_m & \cos \mathcal{G}_m \end{bmatrix} \begin{bmatrix} x(i) \\ y(i) \end{bmatrix} \quad (28)$$

where $\mathcal{G}_m = \Delta_{acc} \frac{2\pi}{2^{16}}$, and Δ_{acc} is an integer number.

(29)

For convergence, the input angle of the scale-free CORDIC algorithm is restricted as follows:

$$\mathcal{G}_m < \sum_{i=4}^{w-1} 2^{-i} \cong \frac{1}{8} \quad (30)$$

From the above two equations, we have

$$\Delta_{acc} = \frac{2^{16}}{2\pi} \cdot \mathcal{G}_m < 1304 \quad (31)$$

The architecture for the sine/cosine generator is shown in Fig. 5. In which three modified scaling-free CORDIC arithmetic units (MCORDIC-Type A) and

two modified scaling-free radix-8 CORDIC arithmetic units (MCORDIC-Type B) are used. According to equation (23).

The chip is synthesized by the TSMC 0.18 μm 1P6M CMOS cell libraries [14]. The layout view of the proposed DDFS is shown in Figure 6. The core size obtained by the Synopsys[®] design analyzer is $612 \times 612 \mu\text{m}^2$. The power consumption obtained by the PrimePower[®] is 6.05 mW with a clock rate of 100MHz at 1.8V. The tuning latency is 8 clock cycles. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

4 Performance Analysis of the Proposed DDFS

The number of correcting points versus the SFDRs with different (F_s/F_o) 's in the proposed DDFS is shown in Fig. 7. Due to trade-off between hardware cost and system performance, the correcting circuit with 16 points is implemented in the proposed DDFS. Thus, the SFDR of the proposed DDFS is more than 84.4 dBc. Table 1 shows various comparisons of the proposed DDFS with other methods in [6] and [10]. As one can see, the proposed DDFS is superior in terms of SFDR, hardware cost, and power consumption.

5 Conclusion

The hybrid CORDIC-based multiplier-less DDFS architecture with small ROM and pipelined data path has been implemented. A SoC designed by 1P6M CMOS has been emulated on Xilinx XC2V6000 FPGA. For 16-bit DDFS, the SFDR of sine and cosine using the proposed architecture are more than 84.4 dBc. Simulation results show that the hybrid CORDIC-based approach is superior to the traditional approach to the design and implementation of DDFS, in terms of SFDR, power consumption, and hardware cost. The 16-bit DDFS is a reusable IP, which can be implemented in various processes with efficient uses of hardware resources for trade-offs of performance, area, and power consumption.

References:

- [1] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital frequency synthesis," *IEEE Proceedings of the Frequency Control Symposium*, June 5-7 1996, pp.942-950.

[2] S. C. Yi, K. T. Lee, J. J. Chen, C. H. Lin, "A low power efficient direct digital frequency synthesizer based on new two-level lookup table," *IEEE Canadian Conference on Electrical and Computer Engineering* 2006, May 2006, pp.963-966.

[3] D. A. Sunderland, R. A. Srauch, S. S. Wharfield, H. T. Peterson, C. R. Cole, "CMOS/SOS frequency synthesizer LSI circuit for spread spectrum communications," *IEEE Journal of Solid-State Circuits*, Vol.19, No.4, August 1984, pp.497-506.

[4] H. T. Nicholas, H. Samueli, B. Kim, "The optimization of direct digital frequency synthesizer performance in the presence of finite word length effects," *IEEE 42nd Annual Frequency Control Symposium*, June 1-3 1988, pp.357-363.

[5] D. D. Caro, E. Napoli, A. G. M. Strollo, "Direct digital frequency synthesizers with polynomial hyperfolding technique," *IEEE Transactions of Circuits and Systems-II: Express Briefs*, Vol.51, No.7, July 2004, pp.337-344.

[6] D. Fu, A. N. Willson, Jr. "A high-speed processor for digital sine/cosine generation and angle rotation" in *Proc. 32nd Asilomar Conf. Signals, Systems and Computers*, Vol.1 1998, pp.177-181

[7] A. Torosyan, D. Fu, A. N. Willson, Jr, "A 300-MHz quadrature direct digital synthesizer/mixer in 0.25- μm CMOS" *IEEE Journal of Solid-State Circuits*, Volume 38, Issue 6, June 2003 pp. 875 - 887

[8] A. Madisetti, A. Y. Kwentus, A. N. Willson Jr, "A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range," *IEEE Journal of Solid-State Circuits*, Vol.34, No.8, August 1999, pp.1034-1043.

[9] E. Grayver, B. Daneshrad, "Direct digital synthesis using a modified CORDIC," *IEEE International Symposium on Circuits and Systems (ISCAS '98)*. Vol.5, May 1998, pp.241-244.

[10] C. Y. Kang, E. E. Swartzlander Jr., "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.53, No.5, May 2006, pp.1035-1044.

[11] T. Y. Sung, H. C. Hsin, "Design and simulation of reusable IP CORDIC core for special-purpose processors," *IET Computers & Digital Techniques*, Vol.1, No.5, Sept. 2007, pp.581-589.

[12] Y. H. Hu, S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Transactions on Computers*, Vol.42, No.1, January 1993, pp.99-102.

[13] T. B. Juang, S. F. Hsiao, M. Y. Tsai, "Para-CORDIC: parallel CORDIC rotation algorithm," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol.51, No.8, August 2004, pp.1515-1524.

[14] "TSMC 0.18 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.

Table I Comparison with previous works

CORDIC Based DDFS	Madisetti [8] 1999	Swartzlander [10] 2006	This work 2009
Process (μm)	1.0	0.13	0.18
Core Area (mm^2)	0.306	0.35	0.375
Maximum Sampling Rate (MHz)	80.4	1018	100
Power Consumption (mw)	40.602	350	6.056
Power Consumption (mw/MHz)	0.505	0.343	0.06
SFDR (dB_c)	81	90	84.4
Output Resolution (bits)	16	16	16
Tuning Latency (clock cycles)	16	--	8

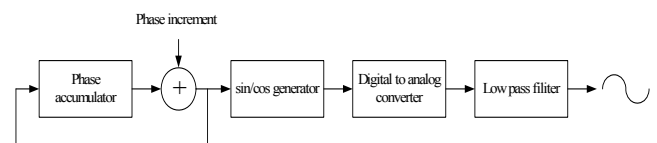


Fig. 1 The conventional DDFS architecture

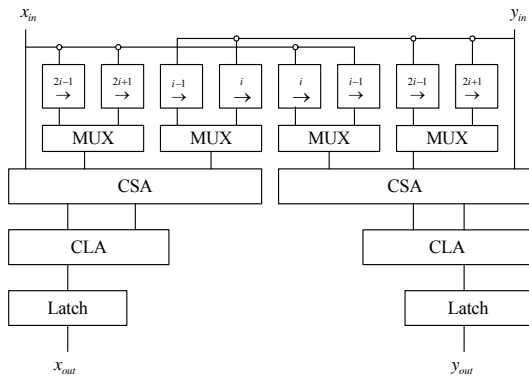


Fig. 2 The proposed architecture of modified scaling-free CORDIC arithmetic for computing θ_H (MCORDIC-Type A)

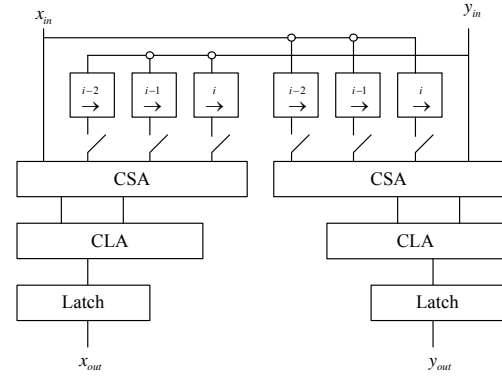


Fig. 3 The architecture of modified scaling-free radix-8 CORDIC arithmetic for computing θ_L (MCORDIC-Type B)

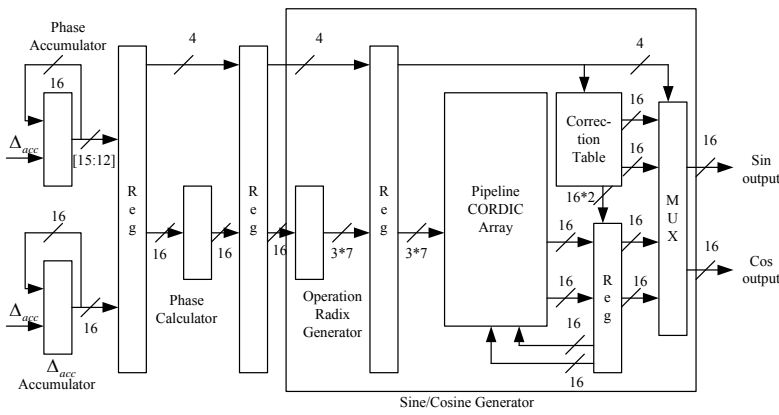


Fig. 4 The 16-bit DDFS architecture

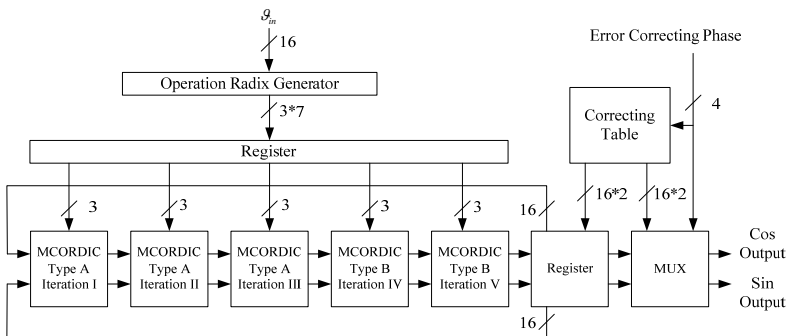


Fig. 5 The architecture of sine/cosine generator (The θ_{in} is an accumulated angle)

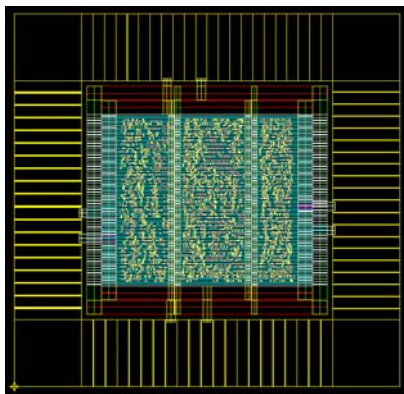


Fig. 6 The layout view of the proposed DDFS

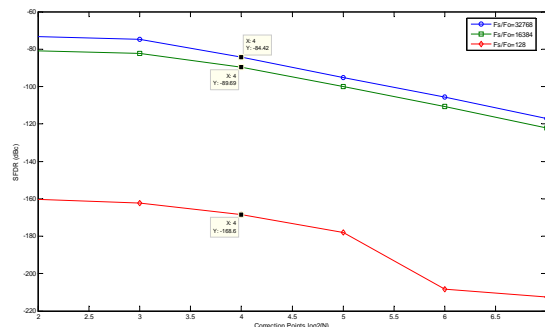


Fig. 7 Plot of the number of correcting points versus SFDRs with different (F_s / F_o) 's

A Simple Rate Distortion Estimation for Embedded Block Coding

HSI-CHIN HSIN

Department of Computer Science and
Information Engineering
National United University
1, Lien-Da, Miao-Li, 36003, Taiwan
hsin@nuu.edu.tw

TZE-YUN SUNG

Department of Microelectronics
Engineering
Chung Hua University
707 Wu-Fu, Hsinchu 30012, Taiwan
bobsung@chu.edu.tw

Abstract: - The embedded block coding with optimized truncation (EBCOT) algorithm has been adopted by the JPEG2000 standard. In which, the post compression rate distortion (PCRD) optimization needs a large memory space to store all the code streams of code blocks; however, code blocks of less importance might not be used for the optimal decoded image at a bit rate. To avoid waste of computational power and memory space, a simple context based rate distortion estimation (CBRDE) is proposed to arrange the scanning order of code blocks in an adaptive manner. CBRDE is based on the MQ table, which is available at both encoder and decoder. Thus, there is no need to store and transmit the rate distortion information of code blocks. Experimental results show that the rate distortion curves are almost convex; this demonstrates the benefit of the proposed CBRDE.

Key-Words: - image coding; wavelet transform; JPEG2000; EBCOT; CBRDE

1 Introduction

Wavelet transform offers many desirable properties, e.g. progressive transmission and embedded coding, which are suitable for image coding [1]-[3]. In general, embedded code streams of an image are organized in decreasing order of information importance. The embedded block coding with optimized truncation (EBCOT) algorithm has been adopted by JPEG2000 [4], which is preferable to the JPEG standard at the cost of increasing complexity [5]. For real-time applications, dedicated hardware is required to implement JPEG2000.

In EBCOT, the tier-1 algorithm applies embedded block coding with an arithmetic coder to each code block of the transform coefficients; the tier-2 algorithm performs rate distortion control by the post compression rate distortion (PCRD) algorithm. One of the crucial implementation issues of EBCOT is the design of PCRD, which requires a large amount of memory space to store all the code streams of code blocks with their respective rate distortion information. As all the code blocks of an image must be processed before PCRD, however, some code blocks may not be needed to reconstruct the optimal decoded image at a bit rate; this surely leads to waste of computational power. In [6], Fang et al. proposed a precompression rate distortion optimization algorithm to reduce the memory space by ignoring the unused code streams.

In this paper, we propose a context based rate distortion estimation (CBRDE) to arrange the scanning order of code blocks so that the available coding bits can be allocated for the most significant

code block. The proposed CBRDE is based on the MQ table of EBCOT, which is available at both encoder and decoder. As a result, the scanning order of code blocks can be also obtained at decoder, and therefore transmission of the contributions of each code block can be omitted. The remainder of this paper proceeds as follows. In Section 2, wavelet transform and EBCOT are reviewed briefly. Section 3 presents the proposed CBRDE algorithm to arrange the code blocks of an image in an adaptive manner. Experimental results are given in Section 4, and conclusion can be found in Section 5.

2 Wavelet Transform and EBCOT

Wavelet transform provides many desirable properties such as multiresolution analysis, subband decomposition with orientation selectivity, joint space-spatial frequency localization, self similarity across subbands of the same orientation, and energy clustering within each subband. In this section, wavelet transform and the EBCOT algorithm are reviewed briefly.

2.1 Wavelet transform

The wavelet transform of a signal, $S_\ell(n)$, at resolution ℓ is as follows.

$$\begin{aligned} S_{\ell+1}(n) &= \sum_k S_\ell(k) h(2n-k) \\ D_{\ell+1}(n) &= \sum_k S_\ell(k) g(2n-k) \end{aligned} \quad (1)$$

where $S_{\ell+1}(n)$ is the approximation at the next coarser resolution $\ell+1$, $D_{\ell+1}(n)$ is the detail between resolutions ℓ and $\ell+1$, $h(n) = \langle \phi, \phi_{-1,-n} \rangle$, $g(n) = \langle \psi, \phi_{-1,-n} \rangle$, $\langle \cdot, \cdot \rangle$ is an inner product operator, ψ is a valid (mother) wavelet, ϕ is the corresponding scaling function and $\phi_{-1,-n}(x) = 2^{-1/2} \phi(2^{-1}x - n)$. $S_{\ell}(n)$ can be exactly reconstructed from $S_{\ell+1}(n)$ and $D_{\ell+1}(n)$ by using following inverse wavelet transform.

$$S_{\ell}(n) = \sum_k S_{\ell+1}(k) \tilde{h}(n-2k) + \sum_k D_{\ell+1}(k) \tilde{g}(n-2k) \quad (2)$$

where $\tilde{h}(n) = h(-n)$ and $\tilde{g}(n) = g(-n)$.

For image applications, 2-D wavelet transform can be obtained by the tensor product of two 1-D wavelet transforms. Similarly, 2-D inverse wavelet transform can be obtained by the tensor product of two 1-D inverse wavelet transforms.

2.2 The EBCOT algorithm

In many cases, the DWT-based JPEG2000 standard outperforms the DCT-based JPEG standard. The idea behind the heart of JPEG2000 known as the EBCOT algorithm is to exploit energy clustering of wavelet coefficients. EBCOT is a two-tier algorithm; tier-1 performs bit-plane coding (BPC) followed by arithmetic coding (AC); tier-2 aims for post compression rate distortion optimization. The quantized wavelet coefficients of an image are coded by a context-based arithmetic coder known as the MQ coder, in which the probability models are stored in the MQ table. Figure 1 depicts block diagram of the JPEG2000 encoder.

In EBCOT, each BPC is further divided into three coding passes, namely significance propagation pass, magnitude refinement pass, and cleanup pass. Four primitive coding operations, namely significance coding operation, sign coding operation, magnitude refinement coding operation, and cleanup coding operation are performed in the corresponding coding passes. For a transform coefficient that is currently insignificant, if any of the 8 neighboring transform coefficients are already significant, it is coded in the significance propagation pass using the significance coding operation; otherwise, it is coded in the cleanup pass using the cleanup coding operation; if this coefficient becomes significant, the sign is coded immediately using the sign coding operation. In the magnitude refinement pass, the

magnitude of the significant transform coefficients that have been found in previous coding passes is updated using the magnitude refinement coding operation. The output bit streams of coding passes can be further coded by using an arithmetic coding engine to improve the compression performances at the cost of computational complexity. Based on the current status of the 8 neighboring transform coefficients, JPEG2000 defines 18 context labels, and stores their respective probability modes in the MQ table. More precisely, 10 context labels are defined for the significance coding operation and the cleanup coding operation, 5 context labels for the sign coding operation, and 3 context labels for the magnitude refinement coding operation.

3 The CBRDE Algorithm

In JPEG2000, a large image is first divided into rectangular sub-images called tiles; each tile is decomposed into subbands by wavelet transform; every subband is partitioned further into small blocks called code blocks, which are quantized to form bit-planes and then coded by EBCOT from the most significant bit-plane to the least significant bit-plane. For each bit-plane, all the code blocks of an image must be processed in the first tier of EBCOT before proceeding with the application of the PCRD algorithm in the second tier. As code blocks of less importance are not needed for the optimal decoded image at a given bit rate, waste of computational power and memory space might result. In this section, an efficient scheme is proposed for the scanning order of code blocks such that waste of computational power and memory space can be reduced.

3.1 Adaptive scanning order

Recall that the code blocks of an image are coded bit-plane by bit-plane, from most to least significant, and the output bitstream can be truncated at an intermediate point between bit-planes; it raises the following interesting questions regarding the scanning order of code blocks. For each bit-plane, is there an adaptive scanning order such that the available coding bits can be allocated for the most significant code block? Is there a common piece of information available at both encoder and decoder, based on which code blocks can be arranged adaptively in decreasing order of significance? If so, there is no need to store all the code streams of code blocks and transmit the scanning order from encoder

to decoder. Is there any simple way to solve the above-mentioned questions?

As one might expect, the first question can be solved effectively by using the rate distortion curve of code blocks. The optimal scanning strategy turns out to take the code block with the steepest rate distortion slope as the first to be coded. In other words, the available coding bits should be first allocated for the code block with the largest amount of distortion decrease per coding bit. Take the second question into consideration; estimated rate distortion slope is preferable to the true one for arranging the scanning order of code blocks. Since the MQ table of probabilities of more probable symbol (MPS) and less probable symbol (LPS) is available at both encoder and decoder, an efficient, context based rate distortion estimation (CBRDE) is therefore proposed to estimate the rate distortion slope of code blocks, which is given in the following subsection.

3.2 Proposed CBRDE

For each bit-plane, code blocks with significant rates of distortion decrease per coding bit should be coded as early as possible. In order to avoid transmitting the scanning order of code blocks from encoder to decoder, the estimated rate distortion slope of a code block is thus adopted, which is given by

$$S = \frac{E[\Delta D]}{E[\Delta R]} \quad (3)$$

where ΔD and ΔR denote the amount of distortion decrease and the number of coding bits, respectively, and $E[\cdot]$ is an expectation operator. Though most of correlation between images pixels can be removed by wavelet transform, there may still be residual correlation between neighboring wavelet coefficients. To incorporate with the framework of EBCOT, the proposed CBRDE for code blocks is based on the MQ table, which is available at both encoder and decoder. Specifically, let B_i denote the i^{th} code block (in wavelet domain), $b_{ij}(m, n)$ the j^{th} significant bit of a coefficient at position: (m, n) in B_i , and $B_{ij} = \bigcup_{m,n} \{b_{ij}(m, n)\}$. The proposed CBRDE for B_{ij} is as follows.

$$S_{ij} = \frac{\sum_{m,n} \text{prob}(b_{ij}(m, n) = 1)}{\sum_{m,n} H(b_{ij}(m, n))} \quad (4)$$

where $H(b_{ij}(m, n))$ denotes the entropy of $b_{ij}(m, n)$ given by

$$H(b) = -\text{prob}(b = 1) \cdot \log_2(\text{prob}(b = 1)) - \text{prob}(b = 0) \cdot \log_2(\text{prob}(b = 0)) \quad (5)$$

The numerator of equation (4) is proportional to the amount of distortion decrease; and the denominator is proportional to the number of coding bits. It is noted that probability models of $b_{ij}(m, n)$ can be obtained directly from the MQ table, and therefore the computational complexity is nothing but one lookup table operation.

Take the commonly used Lena image as an example, the true distortion: ΔD and coding bits: ΔR , and their respective estimates $E[\Delta D]$ and $E[\Delta R]$ (for a given bit-plane) are shown in Fig. 2. As one can see, the nearly proportional relationship between the horizontal axis (true values) and the vertical axis (estimated values) demonstrates the potential of the proposed CBRDE.

3.3 Embedded block coding with CBRDE

Figure 3 depicts the proposed image encoder using embedded block coding (EBC) with adaptive block ordering (ABO). In which, the EBC algorithm is the same as the first tier of EBCOT; however, ABO with CBRDE replaces the second tier of EBCOT, i.e. PCRD. With the adaptation of the MQ table, the code blocks of an image can be arranged according to their respective estimated rate distortion slopes and then coded in an adaptive manner. Figure 4 depicts the image decoder. Recall that the rate distortion slopes of code blocks can be also estimated at decoder. Thus, the overhead of transmitting the scanning order of code blocks from encoder to decoder is unnecessary; this is beneficial to the communication applications.

4 Experimental Results

The proposed CBRDE for arranging the scanning order of code blocks is valuated on grayscale images. The test images, namely Barbara and Fingerprint are shown in Fig. 5. The biorthogonal Daubechies wavelet with 9/7-tap filters is used. The compression rate is measured in bits per pixel (bpp). The distortion is defined as mean squared error (MSE). The compression rates and MSE values are used to plot the rate distortion curves. To avoid the overhead of transmitting the contributions of code blocks, the second tier of EBCOT, i.e. PCRD is replaced by the CBRDE-based ABO. For comparison, as most of images' energies are concentrated in the low frequency subbands, the first tier of EBCOT incorporated with a fixed scanning order starting from the lowest frequency subband is evaluated.

More specifically, the wavelet subbands of an image are coded in a zigzag order from the lowest frequency subband to the highest frequency subband; the code blocks within each subband are scanned from left to right, top to bottom.

For images with textures such as Barbara, there are many significant wavelet coefficients in the middle and high frequency subbands. As shown in Fig. 6, the coding performance can be significantly improved at low bit rates. Figure 7 shows the simulation results of Fingerprint image. It is noted that the rate distortion curve obtained by the proposed image coder is almost convex; this demonstrates the benefit of the proposed CBRDE when the embedded code stream of an image is truncated at an intermediate point between bit-planes.

5 Conclusion

Wavelet transform has been adopted by JPEG2000 as the underlying method to decompose an image into subbands with orientation selectivity. It provides many desirable properties, e.g. multiresolution analysis, high correlation across wavelet subbands of the same orientation, and energy clustering within each subband; these properties are suitable for the image compression applications. In EBCOT, all the code blocks of an image are coded to generate a set of code streams with their respective contributions to the decoded image, based on which the optimal code stream can be obtained by concatenating the suitably truncated code streams through the PCRD algorithm. As some code blocks, which are less important, are not needed for the optimal decoded image at a given bit rate, waste of computational power and memory space may result. Furthermore, implementation of the PCRD algorithm is one of the crucial issues. To overcome the above-mentioned problems, a simple context-based rate distortion estimation (CBRDE) is proposed to arrange the scanning order of code blocks in an adaptive manner. To avoid transmitting the side information regarding the scanning order of code blocks from encoder to decoder, the proposed CBRDE is based on the MQ table, which is available at both encoder and decoder. The second tier of EBCOT, i.e. PCRD can therefore be replaced by the CBRDE-based adaptive block ordering (ABO). Experimental results show that the rate distortion curves are almost convex.

Recall that for each code block, the coding procedure proceeds in distinct passes. Thus, the proposed image coding with adaptive block ordering can be extended to image coding with adaptive pass

ordering, however, at the cost of increasing complexity from the implementation point of view.

References:

- [1] J. M. Shapiro, "Embedded Image Coding Using Zero-Trees of Wavelet Coefficients," *IEEE Trans. On Signal Processing*, vol. 40, pp. 3445-3462, 1993.
- [2] A. Said and W. A. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. On Circuits Syst. Video Tech.* vol. 6, pp. 243-250, 1996.
- [3] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, Low Complexity Image Coding With a Set-Partitioning Embedded Block Coder," *IEEE Trans. On Circuits Syst. Video Tech.* vol. 14, pp. 1219-1235, Nov., 2004.
- [4] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Trans. On Image Processing*, vol. 9, pp. 1158-1170, July, 2000.
- [5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, pp. 36-58, September, 2001.
- [6] H.-C. Fang, Y.-W. Chang, T.-C. Wang, C.-T. Huang, and L.-G. Chen, "High-Performance JPEG 2000 Encoder with Rate-Distortion Optimization," *IEEE Trans. On Multimedia*, vol. 8, no. 4, pp. 645-653, August. 2006.

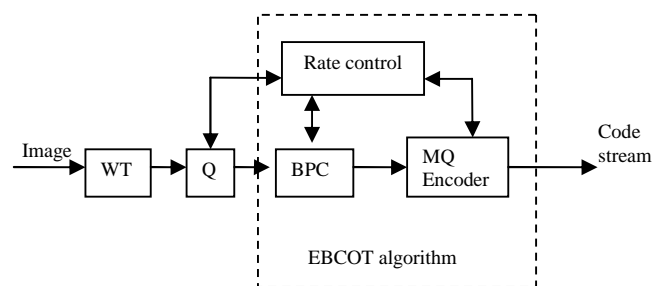
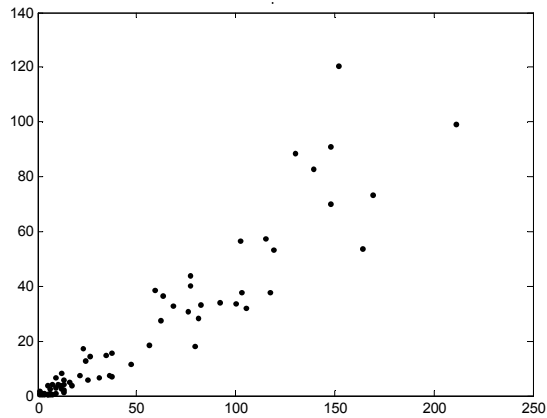
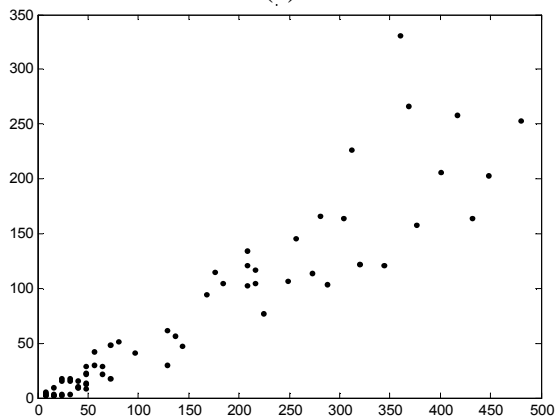


Fig. 1 Block diagram of JPEG2000 Encoder



(a)



(b)

Fig. 2 Performance of CBRDE applied to Lena image; (a) horizontal axis: true ΔD , vertical axis: estimated $E[\Delta D]$; (b) horizontal axis: true ΔR , vertical axis: estimated $E[\Delta R]$.

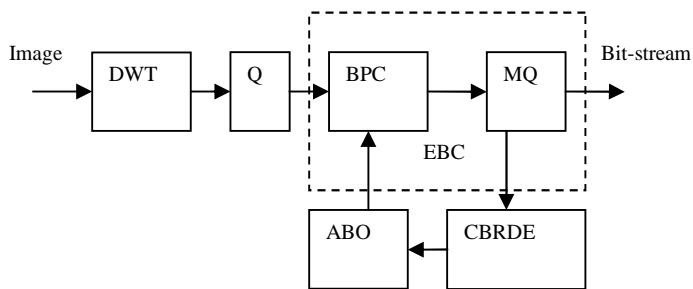


Fig. 3 Proposed image encoder using EBC with ABO

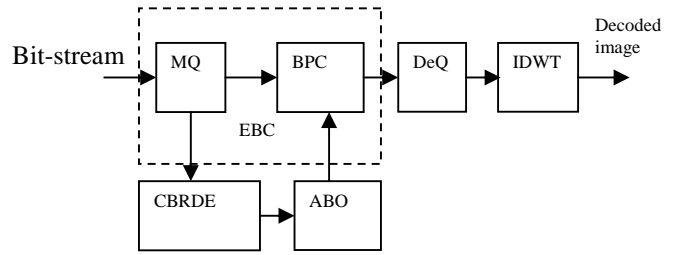


Fig. 4 Proposed image decoder using EBC with ABO



(a)



(b)

Fig. 5 Test images (a) Barbara (b) Fingerprint

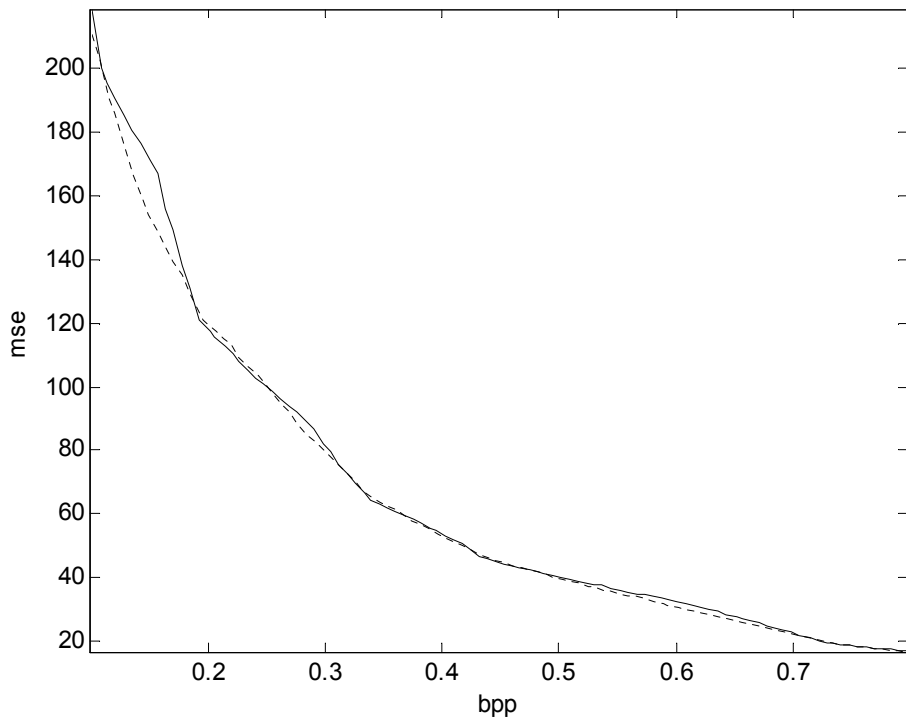


Fig. 6 Rate distortion curves of Barbara image by EBC with the CBRDE-based ABO (dashed line) and EBC with a fixed scan order (solid line); Vertical axis: mean square error (MSE); Horizontal axis: bit rate (bpp).

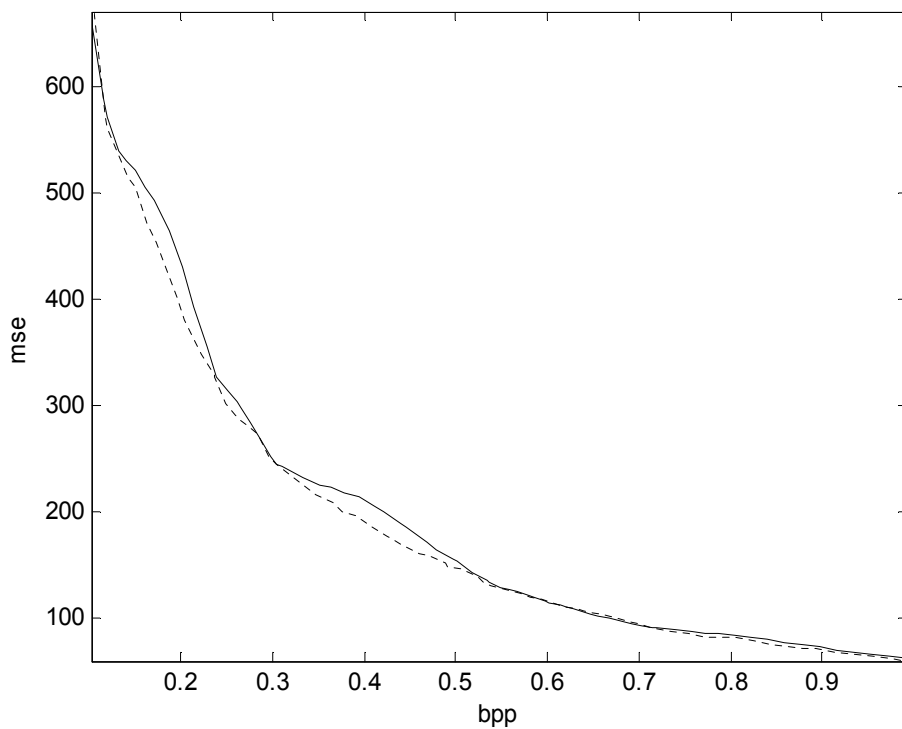


Fig. 7 Rate distortion curves of Fingerprint image by EBC with the CBRDE-based ABO (dashed line) and EBC with a fixed scan order (solid line); Vertical axis: mean square error (MSE); Horizontal axis: bit rate (bpp).

Reconfigurable VLSI Architecture for FFT Processor

Tze-Yun Sung
 Department of Microelectronics
 Engineering
 Chung Hua University
 707, Sec. 2, Wufu Road
 Hsinchu City 300-12, Tawan
 bobsung@chu.edu.tw

Hsi-Chin Hsin
 Department of Computer Science
 and Information Engineering
 National United University
 Miaoli 36003, Taiwan
 hsin@nuu.edu.tw

Abstract: - This paper presents a CORDIC (Coordinate Rotation Digital Computer)-based split-radix fast Fourier transform (FFT) core for OFDM systems, for example, Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL), and Worldwide Interoperability for Microwave Access (WiMAX). High-speed 128/256/512/1024/ 2048/4096/8192-point FFT processors have been implemented by 0.18 (1p6m) at 1.8V, in which all the control signals are generated internally. These FFT processors outperform the conventional ones in terms of both power consumption and core area.

Key-Words: - IP, FFT, split-radix, CORDIC, OFDM.

1 Introduction

High-performance fast Fourier transform (FFT) processor is needed especially for real-time digital signal processing applications. Specifically, the computation of discrete Fourier transform (DFT) ranging from 128 to 8192 points is required for the orthogonal frequency division multiplexer (OFDM) of the following standards: Ultra Wide Band (UWB), Asymmetric Digital Subscriber Line (ADSL), Digital Audio Broadcasting (DAB), Digital Video Broadcasting – Terrestrial (DVB-T), Very High Bitrate DSL (VHDSL) and Worldwide Interoperability for Microwave Access (WiMAX) [1]-[8]. Thompson [9] proposed an efficient VLSI architecture for FFT in 1983. Wold and Despain [10] proposed pipelined and parallel-pipelined FFT for VLSI implementations in 1984. Widhe [11] developed efficient processing elements of FFT in 1997. To reduce the computation complexity, the split-radix 2/4, 2/8, and 2/16 FFT algorithms were proposed in [12]-[15].

As the Booth multiplier is not suitable for hardware implementations of large FFT, we propose the CORDIC-based multiplier. Moreover, we develop a ROM-free twiddle factor generator using simple shifters and adders only [1], which obviates the need to store all the twiddle factors in a large ROM space. As a result, the proposed CORDIC-based split-radix FFT core with the ROM-free twiddle factor generator is suitable for the wireless local area network

(WLAN) applications. In this paper, a high-performance 128/256/512/1024/2048/4096/8192-point FFT processor is presented for the European and Japanese standards. The remainder of this paper proceeds as follows. In Section II, the split-radix 2/8 FFT algorithm and the CORDIC algorithm are reviewed briefly. In Section III, the reusable IP 128-point CORDIC-based split-radix FFT core is proposed. In Section IV, the hardware implementations of FFT processors are described. The performance analysis is presented in Section V. Finally, the conclusion is given in Section VI.

2 Review of Split-Radix FFT and CORDIC Algorithm

2.1 Split-Radix FFT

The idea behind the split-radix FFT algorithm is to compute the even and odd terms of FFT separately. The even term of the split-radix 2/8 FFT algorithm is given by

$$X(2k) = \sum_{n=0}^{N/2-1} (x(n) + x(n + \frac{N}{2}))W_{N/2}^{nk} \quad (1)$$

where $W_{N/2} = e^{-j\frac{2\pi}{N/2}}$ and $k = 0, 1, 2, \dots, (N/2) - 1$.

The odd term is as follows:

The National Science Council of Taiwan, under Grant NSC97-2221-E-216-044, and the Chung Hua University, Hsinchu, Taiwan, under Contract CHU-NSC97-2221-E-216-044 supported this work.

$$\begin{aligned}
 X(8k+l) = & \sum_{n=0}^{N/8-1} \left((x(n) + x(n + \frac{2N}{8})W_4^l + x(n + \frac{4N}{8})W_4^{2l} \right. \\
 & + x(n + \frac{6N}{8})W_4^{3l} + x(n + \frac{N}{8}) + x(n + \frac{3N}{8})W_4^l \quad (2) \\
 & \left. + x(n + \frac{5N}{8})W_4^{2l} + x(n + \frac{7N}{8})W_4^{3l} \right) W_N^{nl} W_{N/8}^{nk}
 \end{aligned}$$

where $k = 0, 1, 2, \dots, (N/8) - 1$ and $l = 1, 3, 5, 7$. The split-radix 2/8 FFT algorithm, which combined with radix-2 and radix-4 proves effective to develop a reusable IP 128-point FFT core.

2.2 CORDIC Algorithm

The CORDIC algorithm in the circular coordinate system is as follows [16].

$$x(i+1) = x(i) - \sigma_i 2^{-i} y(i) \quad (3)$$

$$y(i+1) = y(i) + \sigma_i 2^{-j} x(i) \quad (4)$$

$$z(i+1) = z(i) - \sigma_i \alpha(i) \quad (5)$$

$$\alpha(i) = \tan^{-1} 2^{-i} \quad (6)$$

where $\sigma_i = \text{sign}(z(i))$ with $z(i) \rightarrow 0$ in the rotation mode, and $\sigma_i = -\text{sign}(x(i)) \cdot \text{sign}(y(i))$ with $y(i) \rightarrow 0$ in the vectoring mode. The scale factor:

$k(i)$ is equal to $\sqrt{1 + \sigma_i^2 2^{-2i}}$. After n micro-rotations, the product of the scale factors is given by

$$K_1 = \prod_{i=0}^{n-1} k(i) = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2i}} \quad (7)$$

Notice that CORDIC in the circular coordinate system with rotation mode can be written by

$$\begin{bmatrix} x_n \\ y_n \end{bmatrix} = K_c \begin{bmatrix} \cos z_0 & \sin z_0 \\ -\sin z_0 & \cos z_0 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}, \quad (8)$$

where $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ and $\begin{bmatrix} x_n \\ y_n \end{bmatrix}$ are the input vector and the

output vector, respectively, z_0 is the rotation angle, and K_c is the scale factor. In [1], the circular rotation computation of CORDIC was used for complex multiplication with $e^{-j\theta}$, which is given by

$$\begin{bmatrix} \text{Re}[X'] \\ \text{Im}[X'] \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \text{Re}[X] \\ \text{Im}[X] \end{bmatrix} \quad (9)$$

3 Reusable IP 128-Point CORDIC-Based Split-Radix FFT Core

Figure 1 shows the proposed 128-point CORDIC-based split-radix FFT processor, which can be used as a reusable IP core for various FFT with multiples of 128 points. Notice that the modified split-radix 2/8 FFT butterfly processor and the

ROM-free twiddle factor generator are used. In addition, an internal (128 32-bit) SRAM is used to store the input and output data for hardware efficiency, through the use of the in-place computation algorithm [1]

3.1 CORDIC-Based Split-Radix 2/8 FFT Processor

For the butterfly computation of the proposed CORDIC-based split-radix 2/8 FFT processor, sixteen complex additions, two constant multiplications (CM), and four CORDIC operations are needed, as shown in Figure 2. The CORDIC algorithm has been widely used in various DSP applications because of the hardware simplicity. According to equation (9), the twiddle factor multiplication of FFT can be considered a 2-D vector rotation in the circular coordinate system. Thus, CORDIC in the circular coordinate system with rotation mode is adopted to compute complex multiplications of FFT.

The pipelined CORDIC arithmetic unit can be obtained by decomposing the CORDIC algorithm into a sequence of operational stages. In [17], we derived the error analysis of fixed-point CORDIC arithmetic, based on which, the number of the CORDIC stages can be determined effectively. For example, the number of the CORDIC stages is 12 if the overall relative error of 16-bit CORDIC arithmetic is required to be less than 10^{-3} . The pipelined CORDIC arithmetic unit with 12 stages and an additional pre-scalar stage. In which, the pre-calculated scaling factor $K_c \approx 1.64676$ and the Booth binary recoded format leads to 1.101001. The main concern for the design of the CORDIC arithmetic unit is throughput rather than latency. The proposed CORDIC arithmetic unit in terms of gate counts is less than 4 real multipliers significantly. In addition, the power consumption can be reduced significantly by using the proposed CORDIC arithmetic unit; it has been reduced by 30% according to the report of PrimePower® distributed by Synopsys.

As the twiddle factors: W_8^1 and W_8^3 are equal to $\frac{\sqrt{2}}{2}(1-j)$ and $-\frac{\sqrt{2}}{2}(1+j)$, respectively, a complex number, say $(a+bj)$, times W_8^1 or W_8^3 can be written by

$$(a+bj) \times \left(\frac{\sqrt{2}}{2}(1-j) \right) = \frac{\sqrt{2}}{2} ((a+b) + j(-a+b)) \quad (10)$$

$$(a+bj) \times \left(-\frac{\sqrt{2}}{2}(1+j) \right) = -\frac{\sqrt{2}}{2} ((a-b) + j(a+b)) \quad (11)$$

where $\frac{\sqrt{2}}{2}$ can be represented as $1.0\bar{1}0\bar{1}010$ using the Booth binary recoded form (BBRF). Thus, the CM unit can be implemented by using simple adders and shifters only. Figure 3 shows the pipelined CM architecture, which uses three subtractions/additions and therefore improves on the computation speed significantly.

Based on the above-mentioned CORDIC arithmetic unit and CM unit, the computational circuit and hardware architecture of the CORDIC-based split-radix 2/8 FFT butterfly computation are realized. As one can see, the pipelined CORDIC arithmetic unit aims at increasing the throughput of complex multiplications..

3.2 ROM-Free Twiddle Factor Generator

In the conventional FFT processor, a large ROM space is needed to store all the twiddle factors. To reduce the chip area, a twiddle factor generator is thus proposed. Figure 4 shows the ROM-free twiddle factor generator using simple adders and shifters for 128-point FFT. In which, the 16-bit accumulator is to generate the value $2n\pi$ for each index n ; $n = 2^{\log_2 N - 3} - 1$, the 16-bit shifter is to divide $2n\pi$ by N , and the 16-bit shifter/adder is to produce the twiddle factors: θ_N^{1n} , θ_N^{3n} , θ_N^{5n} and θ_N^{7n} . By using the twiddle factor generator, the chip area and power consumption can be reduced significantly at the cost of an additional logic circuit. Table 1 shows the gate counts of the full-ROM storing all the twiddle factors, the CORDIC twiddle factor generator [1] and the ROM-free twiddle factor generator.

4 Implementation of FFT Processors

The 128/256/512/1024/2048/4096/8192-point FFT processors. In which, the radix-2 and split-radix 2/4 butterfly processors [1] using the pipelined CORDIC arithmetic units and twiddle factor generators are implemented; and moreover, two memory banks (4096/2048/1024/512/256/0 \times 32-bit and 8192/4096/2048/1024/512/256/128 \times 32-bit) are allocated for increased efficiency by using the in-place computation algorithm [1]. Hardware architecture is shown in Figure 5.

The hardware code written in Verilog[®] is running on a workstation with the ModelSim[®] simulation tool and Synopsys[®] synthesis tool (design compiler). The chips are synthesized by the TSMC 0.18 μm 1p6m CMOS cell libraries [18]. The physical circuit is synthesized by the Astro[®] tool. The circuits are evaluated by DRC, LVS and PVS [19].

The layout view of the 8192-point FFT processor is

shown in Figure 6. The core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point and 8192-point FFT processors are shown in Table 2. All the control signals are internally generated on-chip. The chip provides both high throughput and low gate count.

5 Performance Analysis of The Proposed FFT Architecture

FFT processors used to compute 128/256/512/1024/2048/4096/8192-point FFT are composed mainly of the 128-point CORDIC-based split-radix 2/8 FFT core; the computation complexity using a single 128-point FFT core is $O(N/6)$ for N -point FFT. The log-log plot of the CORDIC computations versus the number of FFT points is shown in Figure 7. As one can see, the proposed FFT architecture is able to improve the power consumption and computation speed significantly.

6 Conclusion

This paper presents low-power and high-speed FFT processors based on CORDIC and split-radix techniques for OFDM systems. The architectures are mainly based on a reusable IP 128-point CORDIC-based split-radix FFT core. The pipelined CORDIC arithmetic unit is used to compute the complex multiplications involved in FFT, and moreover the required twiddle factors are obtained by using the proposed ROM-free twiddle factor generator rather than storing them in a large ROM space.

The CORDIC-based 128/256/512/1024/2048/4096/8192-point FFT processors have been implemented by 0.18 μm CMOS, which take 395 μs , 176.8 μs , 77.9 μs , 33.6 μs , 14 μs , 5.5 μs and 1.88 μs to compute 8192-point, 4096-point, 2048-point, 1024-point, 512-point, 256-point and 128-point FFT, respectively.

The CORDIC-based FFT processors are designed by using the portable and reusable Verilog[®]. The 128-point FFT core is a reusable intellectual property (IP), which can be implemented in various processes and combined with an efficient use of hardware resources for the trade-offs of performance, area, and power consumption.

References:

- [1] T. Y. Sung, "Memory-efficient and high-speed split-radix FFT/IFFT processor based on

- pipelined CORDIC rotations,” *IEE Proc.-Vis. Image Signal Proc.*, Vol. 153, No. 4, Aug. 2006, pp.405-410.
- [2] J. C. Kuo, C. H. Wen, A. Y. Wu, “Implementation of a programmable 64/spl sim/2048-point FFT/IFFT processor for OFDM-based communication systems,” *Proceedings of the 2003 International Symposium on Circuits and Systems*, Volume 2, 25-28 May 2003 pp.II-121 - II-124.
- [3] L. Xiaojin, Z. Lai, C. J. Cui, “A low power and small area FFT processor for OFDM demodulator,” *IEEE Transactions on Consumer Electronics*, Volume 53, Issue 2, May 2007, pp. 274 – 277.
- [4] J. Lee, H. Lee, S. I. Cho, S. S. Choi, “A high-speed, low-complexity radix-216 FFT processor for MB-OFDM UWB systems,” *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems*, May 2006, pp.
- [5] A. Cortes, I. Velez, J. F. Sevillano, A. Irizar, “An approach to simplify the design of IFFT/FFT cores for OFDM systems,” *IEEE Transactions on Consumer Electronics*, Volume 52, Issue 1, Feb. 2006, pp.26 – 32.
- [6] Y. H. Lee, T. H. Yu, K. K. Huang, A. Y. Wu, “Rapid IP design of variable-length cached-FFT processor for OFDM-based communication systems,” *IEEE Workshop on Signal Processing Systems Design and Implementation*, Oct. 2006 pp.62-65.
- [7] C. L. Wey, W. C. Tang, S. Y. Lin, “Efficient memory-based FFT architectures for digital video broadcasting (DVB-T/H),” *2007 International Symposium on VLSI Design, Automation and Test*, 25-27 April 2007, pp.1-4.
- [8] Y. W. Lin, H. Y. Liu, C. Y. Lee, “A 1-GS/s FFT/IFFT processor for UWB applications,” *IEEE Journal of Solid-State Circuits*, Volume 40, Issue 8, Aug. 2005, pp.1726-1735.
- [9] C. D. Thompson, “Fourier transform in VLSI,” *IEEE Transactions on Computers*, Vol.32, No. 11, 1983, pp.1047-1057.
- [10] E. H. Wold, A. M. Despain, “Pipelined and parallel-pipelined FFT processor for VLSI implementation,” *IEEE Transactions on Computers*, Vol.33, No. 5, 1984, pp.414-426.
- [11] T. Widhe, “Efficient implementation of FFT processing elements,” *Linkoping Studies in Science and Technology*, Thesis No. 619, Linkoping University, Sweden, 1997.
- [12] P. Duhamel, H. Hollmann, “Implementation of "split-radix" FFT algorithms for complex, real, and real symmetric data.” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Volume 10, April 1985, pp.784 – 787.
- [13] A .A. Petrovsky, S. L. Shkredov, “Automatic generation of split-radix 2-4 parallel-pipeline FFT processors: hardware reconfiguration and core optimizations,” *2006 International Symposium on Parallel Computing in Electrical Engineering*, pp.181-186.
- [14] S. Bouguezal, M. O. Ahmad, M. N. S. Swamy, “A new radix-2/8 FFT algorithm for length-q/spl times/2/sup m/ DFTs,” *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Volume 51, Issue 9, 2004, pp.1723- 1732.
- [15] W. C. Yeh, C. W. Jen, “High-speed and low-power split-radix FFT.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 51, Issue 3, March 2003, pp.864 – 874.
- [16] M. D. Ercegovac, T. Lang, “CORDIC algorithm and implementations.” Digital Arithmetic, Morgan Kaufmann Publishers, 2004, Chapter 11.
- [17] T. Y. Sung, H. C. Hsin, “Fixed-point error analysis of CORDIC arithmetic for special-purpose signal processors,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E90-A, No.9, Sep. 2007, pp.2006-2013.
- [18] “TSMC 0.18 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.
- [19] Cadence design systems: [http://www.cadence.com/products /pages/default.aspx](http://www.cadence.com/products/pages/default.aspx).

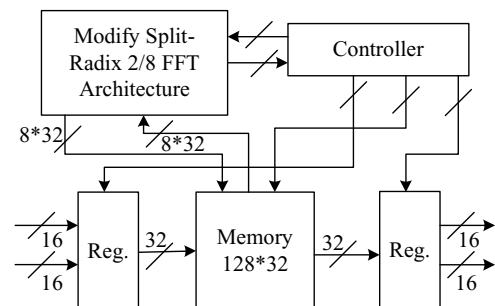


Fig. 1 The proposed 128-point CORDIC- based split-radix FFT processor

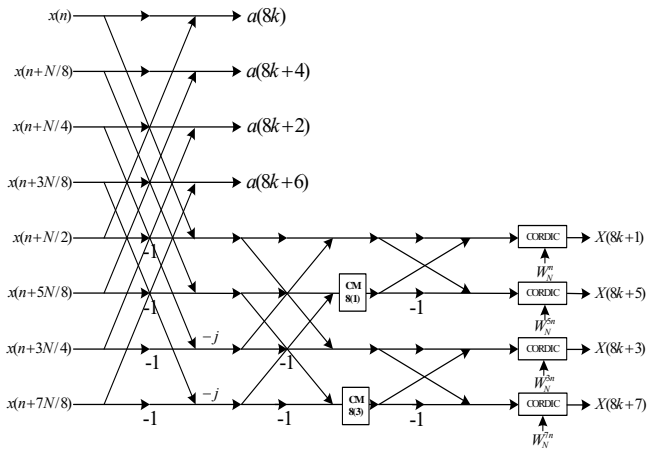


Fig. 2 Data flow of the butterfly computation of the modified split-radix 2/8 FFT

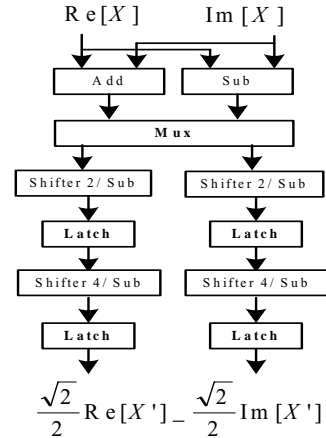


Fig. 3 Constant multiplier (CM) architecture for the modified split-radix 2/8 FFT

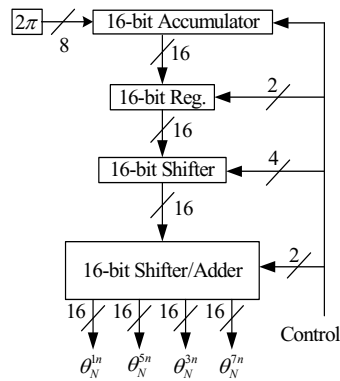


Fig. 4 Proposed ROM-free twiddle factor generator for 128-point FFT

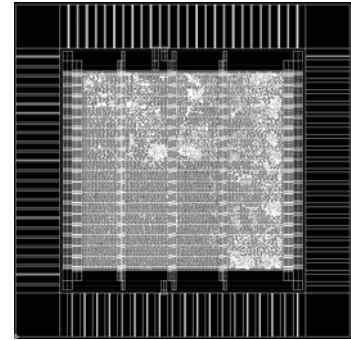


Fig. 6 Layout view of the 8192-point FFT processor

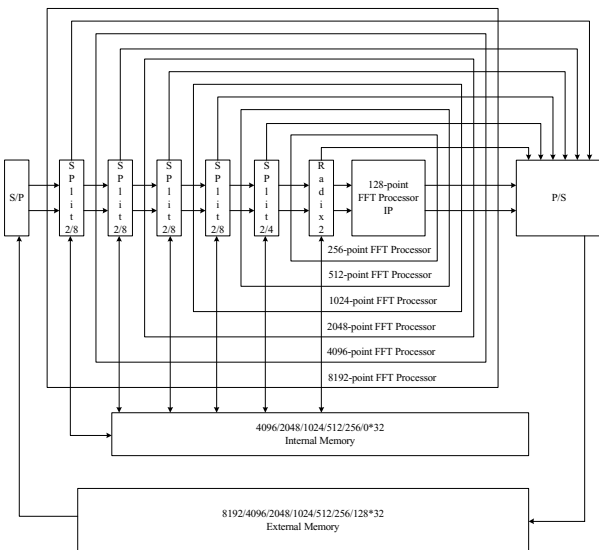


Fig. 5 Hardware architecture of the 128/256/512/1024/2048/4096/8192-point FFT processor

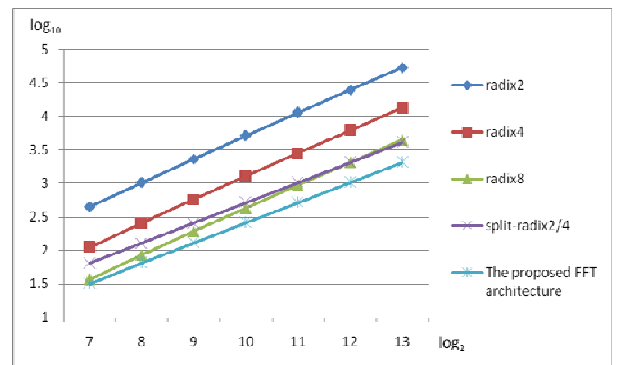


Fig. 7 Log-log plot of the CORDIC computations versus the number of FFT points

Table 1 Hardware requirements of the full-ROM, the CORDIC twiddle factor generator [1], and the ROM-free twiddle factor generator

Full-Twiddle Factor ROM					1bit~1gate
8192-Point ROM					
4K×16 bit					
CORDIC Twiddle Factor Generator (T. Y. Sung, 2006) [1]					
16-bit CORDIC ~18K bit	11-bit Adder ~150 gates	11-bit Shifter ~50 gates	16-bit Shifter ~90 gates	16-bit Adder ~200 gates	
ROM-free Twiddle Factor Generator (Sung, Hsin and Cheng, 2008)					
16-bit Accumulator ~200gates	16-bit Register ~32 gates	16-bit Shifter ~90 gates	16-bit Shifter/Adder ~90×2+200×2 gates		

Table 2 Core areas, power consumptions, clock rates of 128-, 256-, 512-, 1024-, 2048-, 4096- and 8192-point FFT generator

FFT Size	Core Area	Power Consumption	Clock Rate
128-point	2.28mm ²	80mW	200MHz
256-point	2.37mm ²	84mW	200MHz
512-point	2.49mm ²	88mW	200MHz
1024-point	2.62mm ²	94mW	200MHz
2048-point	2.81mm ²	99mW	200MHz
4096-point	3.10mm ²	106mW	200MHz
8192-point	3.62mm ²	117mW	200MHz