# Reconfigurable VLSI Architecture for FFT Processor

**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

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

*Abstract:* - This paper presents a reusable intellectual property (IP) Coordinate Rotation Digital Computer (CORDIC)-based split-radix fast Fourier transform (FFT) core for orthogonal frequency division multiplexer (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). The high-speed 128/256/512/1024/2048/4096/8192-point FFT processors and programmable FFT processor have been implemented by 0.18 $\mu m$ (1$p$6$m$) 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, CORDIC, split-radix, OFDM systems.

## 1 Introduction

High-performance fast Fourier transform (FFT) processor is needed especially for real-time digital signal processing (DSP) 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]-[11]. Thompson [12] proposed an efficient VLSI architecture for FFT in 1983. Wold and Despain [13] proposed pipelined and parallel-pipelined FFT for VLSI implementations in 1984. Widhe [14] 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 [15]-[18].

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 very suitable for the wireless local area network (WLAN) applications.

In this paper, a high-performance 128/256/512/1024/2048/4096/8192-point FFT processors and programmable FFT processor are presented for the European and Japanese standards. The remainder of this paper 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

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

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

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

$$\begin{aligned} X(8k+l) = \sum_{n=0}^{N/8-1} & ((x(n) + x(n+\frac{2N}{8})W_4^l \\ & + x(n+\frac{4N}{8})W_4^{2l} + x(n+\frac{6N}{8})W_4^{-l}) \\ & + (x(n+\frac{N}{8}) + x(n+\frac{3N}{8})W_4^l \\ & + x(n+\frac{5N}{8})W_4^{2l} \\ & + x(n+\frac{7N}{8})W_4^{-l})W_8^{-l})W_N^{nl}W_{N/8}^{nk} \end{aligned} \quad (2)$$

where $k = 0,1,2,....,(N/8)-1$ 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 [19].

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

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

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

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

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

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

## 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 ($128 \times 32$-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 [20], 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 $10^{-3}$. 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. Table 1 shows a comparison between the conventional complex multiplier using 4 real Booth multipliers and the proposed CORDIC arithmetic unit in terms of gate counts. 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

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

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

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 shown in Figure 4, respectively. 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 5 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: $\theta_N^{1n}$, $\theta_N^{3n}$, $\theta_N^{5n}$ and $\theta_N^{7n}$. 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 2 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 Hardware Implementations of FFT Processors by Using IP 128-Point FFT Core

Figure 6 depicts 128/256/512/1024/2048/4096/8192 -point FFT processors; and moreover, two memory banks (4096/2048/1024/512/256/0×32-bit and 8192/4096/2048/1024/512/256/128×32-bit) are allocated for increased efficiency by using the in-place computation algorithm [1]. Hardware architectures of 128/256/512/1024/2048/4096/8192-point FFT processors is shown in Figure 7.

The platform for architecture development and verification has been designed and implemented in order to evaluate the development cost. In which, the 8051 microcontroller reads data from PC via DMA channel and writes the result back to PC by USB 2.0 bus; the Xilinx XC2V6000 FPGA chip [21] implements FFT processors. In addition, the reusable IP CORDIC-based FFT core has been implemented in Matlab® for functional simulations.

The hardware code written in Verilog® is running on a workstation with the modelSim® simulation tool and Synopsys® synthesis tool (design compiler). The chip is synthesized by the TSMC 0.18 $\mu m$ *1p6m* CMOS cell libraries [22]. The physical circuit is synthesized by the Astro® tool. The circuit is evaluated by DRC, LVS and PVS [23].

The layout views, core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point and 8192-point FFT processors and programmable FFT processor are shown in Figure 8. The core areas are obtained by the Synopsys® design analyzer. The power consumptions are obtained by the PrimePower®. All the control signals are internally generated on-chip. The chips provide both high throughput and low gate count. Table 3 shows various comparisons between the proposed FFT architecture and others in [1], [6], [8], [24], and [25].

## 5 Performance Analysis of the Proposed FFT Architecture and Programmable FFT Processor

The proposed 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. By comparison with the CORDIC-based radix-2, radix-4, radix-8 and split-radix 2/4 FFT architectures, the proposed FFT architecture is superior, as shown in Table 4. The plot and log-log plot of the CORDIC computations versus the number of FFT points are shown in Figures 9 and 10, respectively. 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.

CORDIC-based 128/256/512/1024/2048/4096/8192-point FFT processors have been implemented by 0.18 $\mu m$ CMOS, which take 395 $\mu s$, 176.8 $\mu s$, 77.9 $\mu s$, 33.6 $\mu s$, 14 $\mu s$, 5.5 $\mu s$ and 1.88 $\mu s$ 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 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 Procss.*, 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] T. H. Tsai, C. C. Peng, T. M. Chen, "Design of a FFT/IFFT soft IP generator using on OFDM communication system," *WSEAS Transactions on Circuits and Systems*, Vol. 5, no. 8, pp. 1173-1180. Aug. 2006

[10] T. Freyza, S. Hanus, "Hardware implementation of OFDM modulator and demodulator using TMS320C6711 DSK board," *WSEAS Transactions on Circuits and Systems*, Vol. 3, no. 9, pp. 1825-1829. Nov. 2004

[11] X. Yan, Y. Weiyong, H. Chengjun, J. Chuanwen, "Suppression of partial discharge's discrete spectral interference based on spectrum estimation and wavelet packet transform," *WSEAS Transactions on Circuits and Systems*, Vol. 4, no. 11, pp. 1508-1515. Nov. 2005

[12] C. D. Thompson, "Fourier transform in VLSI," *IEEE Transactions on Computers*, Vol.32, No. 11, 1983, pp.1047-1057.

[13] 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.

[14] T. Widhe, "Efficient implementation of FFT processing elements," *Linkoping Studies in Science and Technology*, Thesis No. 619, Linkoping University, Sweden, 1997.

[15] 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.

[16] 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.

[17] S. Bouguezel, 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.

[18] 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.

[19] M. D. Ercegovac, T. Lang, "CORDIC algorithm and implementations." *Digital Arithmetic, Morgan Kaufmann Publishers*, 2004, Chapter 11.

[20] 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.

[21] Xilinx FPGA products: http://www.xilinx.com/products.

[22] "TSMC 0.18 CMOS Design Libraries and Te

Manufacturing Company, Hsinchu, Taiwan, and National Chip Implementation Center (CIC), National Science Council, Hsinchu, Taiwan, R.O.C., 2006.

[23] Cadence design systems: http://www.cadence.com/products/pages/default.aspx.

[24] H. L. Lin, H. Lin, R. C. Chang, S. W. Chen, C. Y. Liao, C. H. Wu, "A high-speed highly pipelined 2N-point FFT architecture for a dual OFDM processor," *Proceedings of the International Conference on Mixed Design of Integrated Circuits and System*, 22-24 June 2006, pp.627 – 631.

[25] Y. W. Lin, H. Y. Liu, C. Y. Lee, "A dynamic scaling FFT processor for DVB-T applications." *IEEE Journal of Solid-State Circuits*, Volume 39, Issue 11, Nov. 2004, pp.2005-2013.

[26] T. Y. Sung, C. S. Chen, "A parallel-pipelined processor for fast Fourier transform," *Fourth IEEE Asia-Pacific Conference on Advanced System Integration Circuits (AP-ASIC)*, 2004, pp.194-197.

Table 1 Hardware comparison between the pipelined complex multiplier using 4 real Booth multipliers and the proposed pipelined CORDIC arithmetic unit.

| Arithmetic unit | 16-bit Pipelined Complex multiplier (4-real Booth multiplier) | Pipelined CORDIC arithmetic unit (16-bit operand) |
|---|---|---|
| Gate counts | ~40 000 | ~20 700 |

Table 2 Hardware requirements of the full-ROM storing all the twiddle factors, the CORDIC twiddle factor generator [1], and the ROM-free twiddle factor generator

| Full-Twiddle Factor ROM | | | | 1Bit~1gate |
|---|---|---|---|---|
| 8192-Point ROM<br>4K×16 bit | | | | |
| CORDIC Twiddle Factor Generator | | (T. Y. Sung, 2006) [1] | | |
| 16-bit CORDIC<br>~ 18K bit | 11-bit Adder<br>~ 150 gates | 11-bit Shifter<br>~ 50 gates | 16-bit Shifter<br>~ 90 gates | 16-bit Adder<br>~ 200 gates |
| ROM-free Twiddle Factor Generator (This Work) | | | | |
| 16-bit Accumulator<br>~ 200gates | 16-bit Register<br>~ 32 gates | 16-bit Shifter<br>~ 90 gates | 16-bit Shifter/Adder<br>$\sim 90\times 2+200\times 2$ gates | |

Table 3 Comparisons between the proposed FFT architecture and others

| Architecture | FFT size | Technology | Word length | Clock rate | Power | Core area |
|---|---|---|---|---|---|---|
| H.L.Lin[21] | 64 | 0.18$\mu m$ 1p6m | 16 bit | 20 MHz | 87mW | 1.59 $mm^2$ |
| Y.W.Lin[8] | 128 | 0.18$\mu m$ 1p6m | 10 bit | 110 MHz | 77.6mW | 3.1 $mm^2$ |
| Y.H.Lee[6] | 2048 | 0.18$\mu m$ 1p6m | 16 bit | 75 MHz | 150mW | 2.1 $mm^2$ |
| T.Y.Sung[1] | 8192 | 0.18$\mu m$ 1p6m | 16 bit | 150 MHz | 350mW | 38.31 $mm^2$ |
| Y.W.Lin[22] | 8192 | 0.18$\mu m$ 1p6m | 11 bit | 20 MHz | 25.2mW | 5.11 $mm^2$ |
| This work | 8192 | 0.18$\mu m$ 1p6m | 16 bit | 200 MHz | 117mW | 3.63 $mm^2$ |

Table 4 Comparison of the computation complexity using various CORDIC-based FFT

| $N$-point FFT (CORDIC-based) | Number of CORDIC computations |
|---|---|
| Radix-2 [1] | $(N/2)\log_2 N$ |
| Radix-4 [1] | $(N/4)\log_4 N$ |
| Radix-8 [23] | $(N/8)\log_8 N$ |
| Split-radix 2/4 [1] | $(N/4)(2-2^{-(\log_2 N-2)})+1$ |
| This work (using a single 128-point FFT core) $N \geq 2^n, n \geq 7$ | $(N/6)$ |

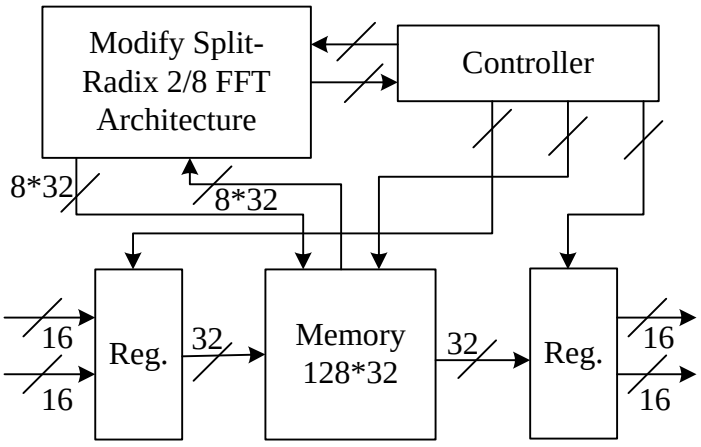


Figure 1 The proposed 128-point CORDIC-based split-radix FFT processor (which can be used as a reusable IP core for various FFT with multiples of 128 points)

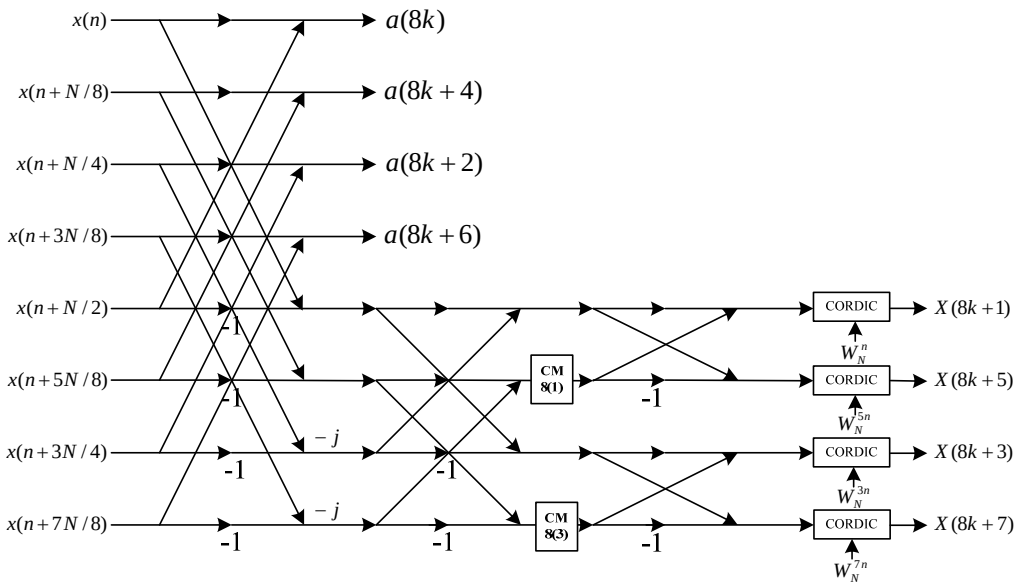


Figure 2 Data flow of the butterfly computation of the modified split-radix 2/8 FFT

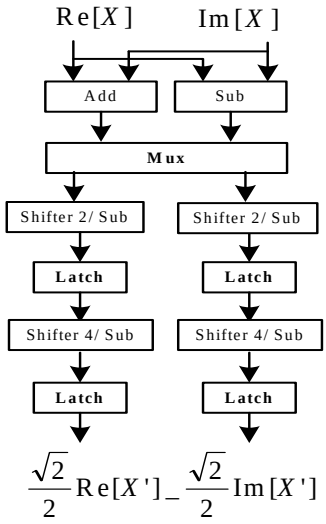


Figure 3 Constant multiplier (CM) architecture for the butterfly computation of the modified split-radix 2/8 FFT

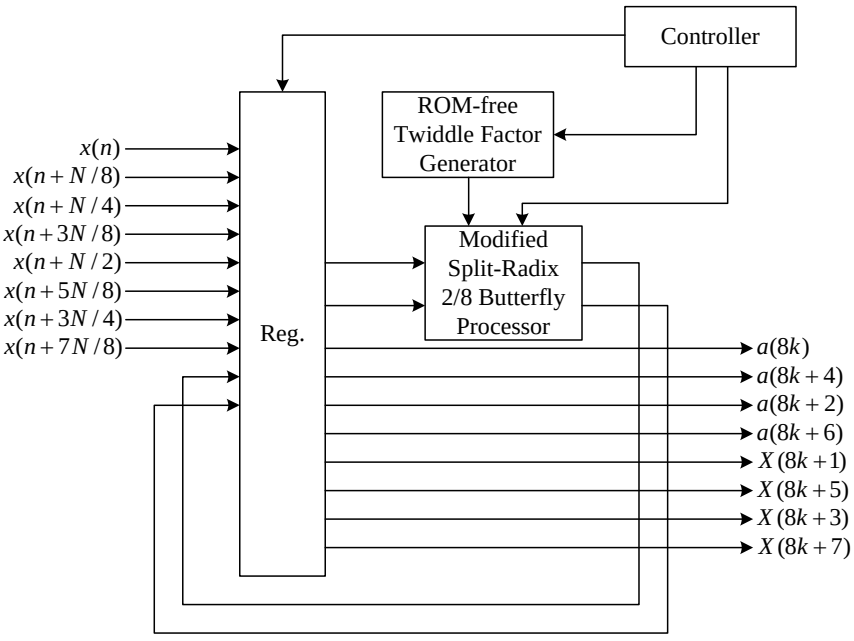


Figure 4 Hardware architecture of the CORDIC-based split-radix 2/8 FFT (Reg.: Registers)

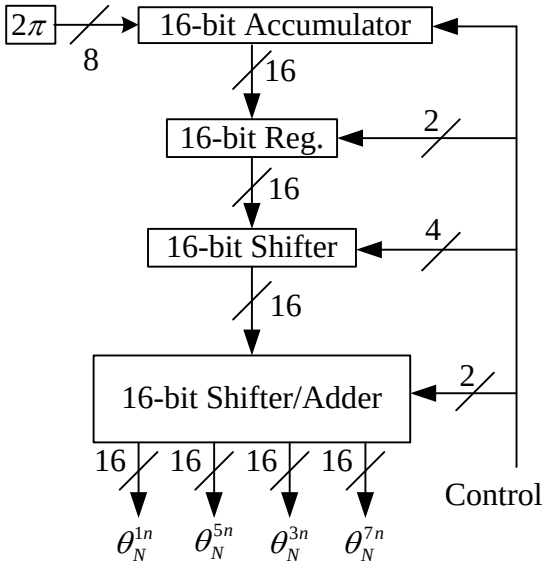


Figure 5 Proposed ROM-free twiddle factor generator for 128-point FFT

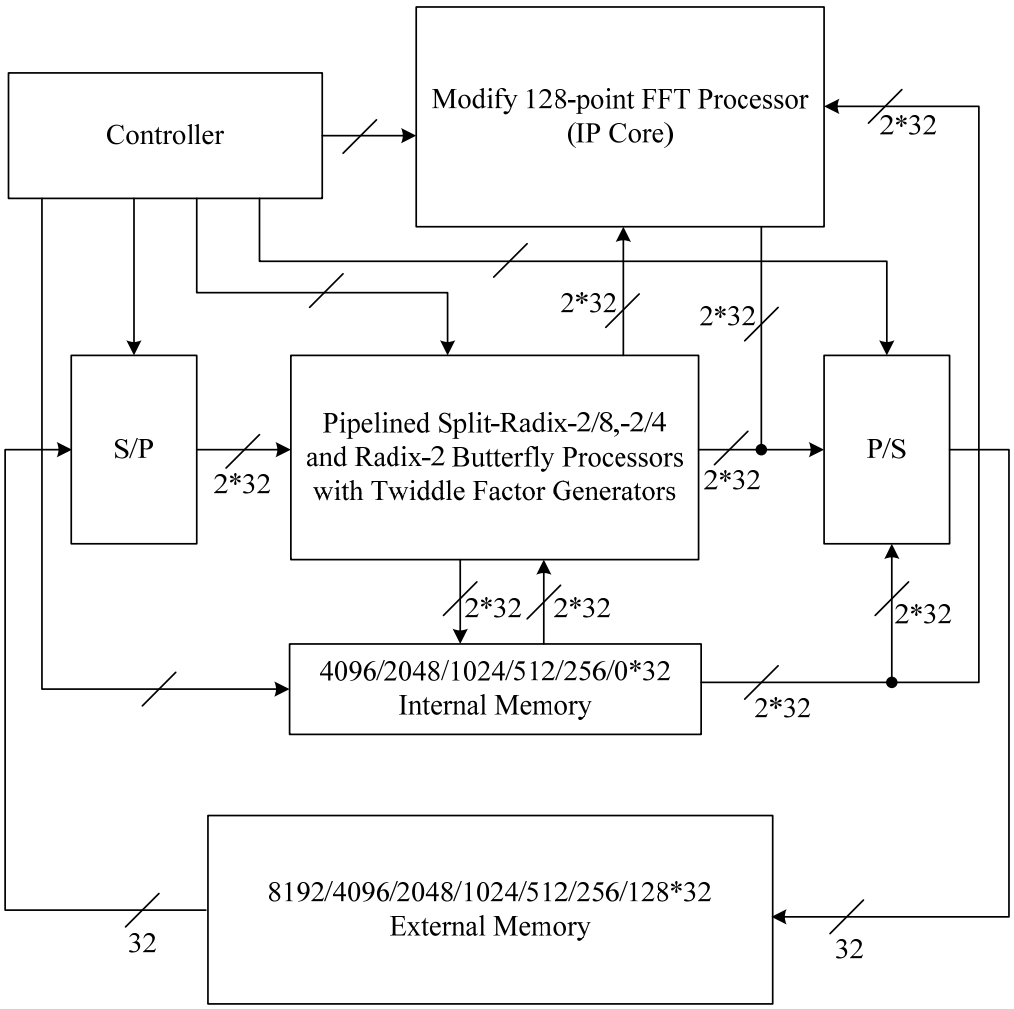


Figure 6 128/256/512/1024/2048/4096/8192-point FFT processors (S/P: serial data to parallel data, P/S: parallel data to serial data)

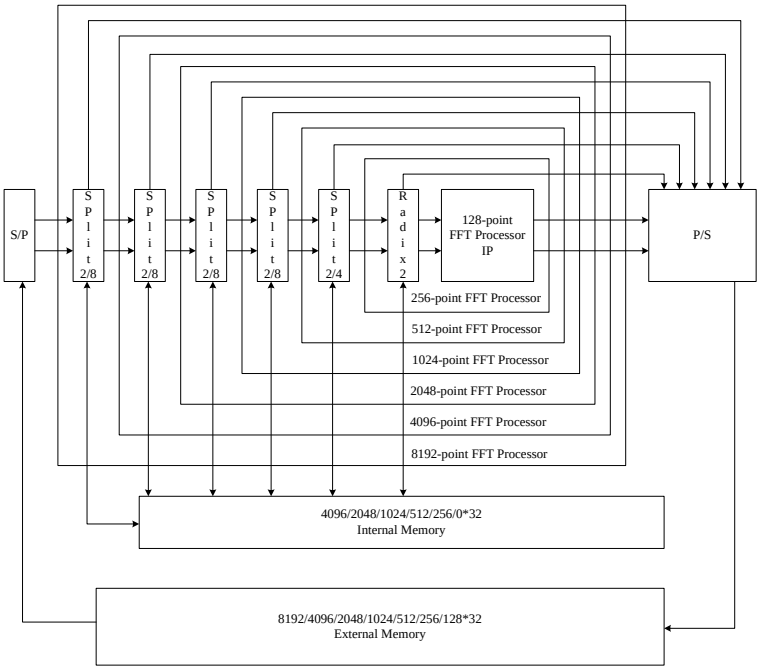


Figure 7 Hardware architectures of 128/256/512/1024/2048/4096/8192-point FFT processors

| FFT Size/Layout View | Core Area | Power Consumption | Clock Rate |
|---|---|---|---|
| 128-point  | $2.28mm^2$ | 80$m$W | 200MHz |
| 256-point  | $2.37mm^2$ | 84$m$W | 200MHz |
| 512-poiint  | $2.49mm^2$ | 88$m$W | 200MHz |
| 1024-point  | $2.62mm^2$ | 94$m$W | 200MHz |
| 2048-point  | $2.81mm^2$ | 99$m$W | 200MHz |
| 4096-point  | $3.10mm^2$ | 106$m$W | 200MHz |
| 8192-point  | $3.62mm^2$ | 117$m$W | 200MHz |
| 128/256/512/1024/2048/4098 Programmable Processor  | $3.65mm^2$ | 117$m$W | 200MHz |

Figure 8 Layout views, core areas, power consumptions, clock rates of 128-point, 256-point, 512-point, 1024-point, 2048-point, 4096-point, 8192-point FFT processors and 28/256/512/1024/2048/4098-point programmable processor

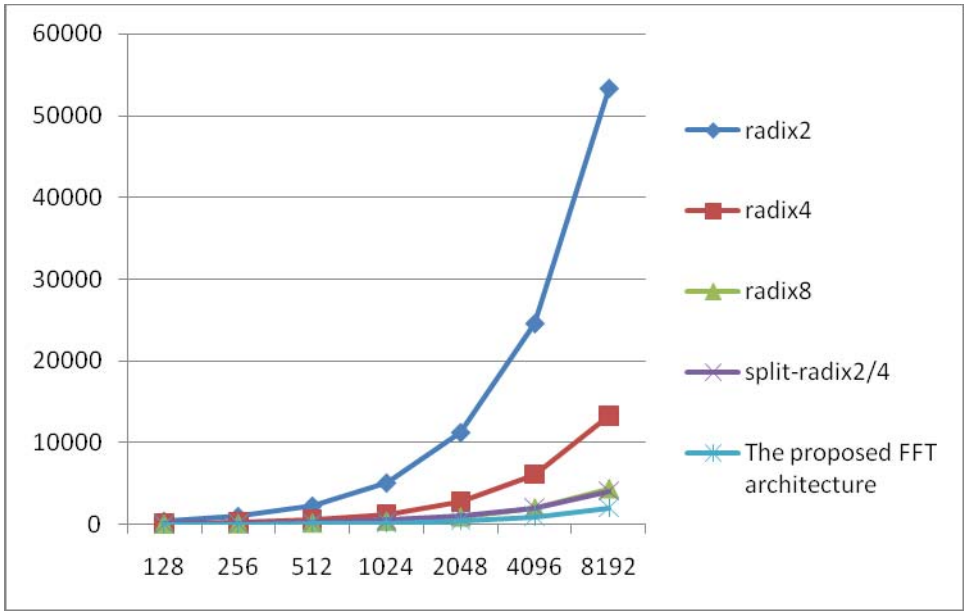


Figure 9 Plot of the CORDIC computations versus the number of FFT points

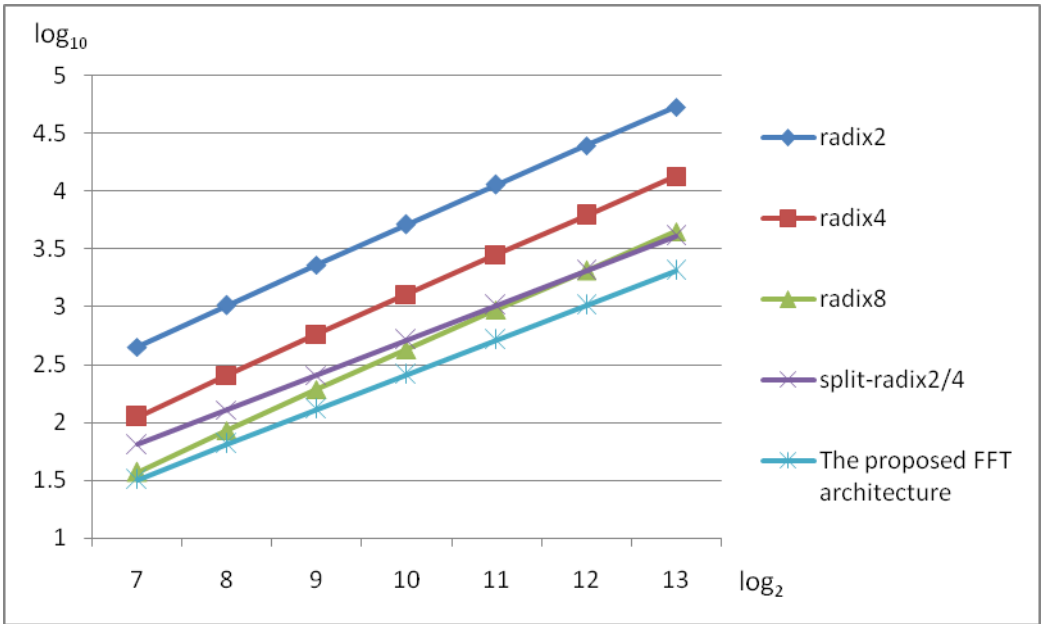


Figure 10 Log-log plot of the CORDIC computations versus the number of FFT points