ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

VLSI Implementation of Pipelined Fast Fourier Transform

K. Indirapriyadarsini[1], S.Kamalakumari[2], G. Prasannakumar[3]
Swarnandhra Engineering College [1&2], Vishnu Institute of Technology [3]
{darsiniprasanna1306[1], Kamalakumari16[2], *godiprasanna[3]}@gmail.com*

Abstract:

Digital Signal Processing (DSP) has become a very important and dynamic research area. Now a day's many integrated circuits dedicated to DSP functions. Unfortunately Existing designs are restricted to a low accuracy and a small sample number. The Fourier transform is widely used in industrial applications as well as in scientific research. The most common use is to transform a function of time into a frequency function. In this paper, we present the efficient implementation of a pipeline FFT. Our design adopts a single-path delay feedback style as the proposed hardware architecture. To eliminate the read-only memories (ROM's) used to store the twiddle factors, the proposed architecture applies a reconfigurable complex multiplier and bit-parallel multipliers to achieve a ROM-less FFT processor, thus consuming lower power than the existing works.

Index Terms: FFT, ROM, complex multiplier.

I. INTRODUCTION

Discrete Fourier transform (DFT) is a very important technique in modern digital signal processing (DSP) and telecommunications, especially for applications in orthogonal frequency demodulation multiplexing (OFDM) systems, such as IEEE 802.11a/g [1], Worldwide Interoperability for Microwave Access (WiMAX) [2], Long Term Evolution(LTE) [3], and Digital Video Broadcasting—Terrestrial(DVB-T) [4]. However, DFT is computational intensive and has a time complexity of $O(N^2)$. The fast Fourier transform (FFT) was proposed by Cooley and Tukey [5] to efficiently reduce the time complexity to O(Nlog 2N), where N denotesthe FFT size.

For hardware implementation, various FFT processors have been proposed [6]. These implementations can be mainlyclassified into memory-based and pipeline architecture styles. Memory-based architecture is widely adopted to design anFFT processor, also known as the single processing element (PE) approach. This deign style is usually composed of amain PE and several memory units, thus the hardware cost and the power consumption are both lower than the other architecture style. However, this kind of architecture style has long latency, low throughput, and cannot be parallelized. On the other hand, the pipeline architecture style can get rid of the disadvantages of the foregoing style, at the cost of anacceptable hardware overhead. Generally, the pipeline FFT processors have two popular design types. One uses single-path delay feedback (SDF) pipeline architecture and the other uses multiple-path delay commentator (MDC) pipeline architecture. The single-path delay feedback (SDF) pipeline FFT [6]-[7] is good in its requiring less memory space (about *N*-1 delay elements) and its multiplicationcomputation utilization being less than 50%, as well as its control unit being easy to design. Such implementations are advantageous to low-power design, especially for applications in portable DSP devices. Based on these reasons, the SDF pipeline FFT is adopted in our work. Our proposed architecture includes a reconfigurable complex constant multiplier and bit-parallel complex multipliers instead of using ROM's to store twiddle factors, which is suited for the power-of-2 radix style of FFT/IFFT processors. In essence, a short version of the present work has been published in [10]. In this paper, a more detailed and completed description of the entire work is provided.The rest of this paper is organized as follows. First, a brief review of the fast Fourier transform is described in Section II. Section III presents our proposed FFT architecture for application in wireless communication systems. The performance evaluation of various FFT architectures is then discussed in Section IV. Finally, concluding remarks are given in Section V.

II. FFT AND IFFT ALGORITHMS

The discrete Fourier transforms (DFT) $X_k$of an *N*-point discrete-time signal $x_n$is defined by:

$$X[k] = \sum_{n=0}^{N-1} x_n W_N^{kn} \qquad 0 \le k \le N-1, \qquad (1)$$

Where the twiddle factor $W_N^{kn} = e^{-\frac{j2\pi kn}{N}}$ denotes N-point primitive root of unity. However, a straightforward implementation of this algorithm is obviously impractical due to the huge hardware

required. Therefore, the fast Fourier transform (FFT) [5] was developed to efficiently speed up its Computation time and significantly reduce the hardware cost. Generally, FFT analyzes an input signal sequence by using decimation-in-frequency (DIF) or decimation-in-time (DIT) decomposition to construct an efficiently computational signal-flow graph (SFG). Here, our work employs a DIFdecomposition because it matches the manipulation manner of single-path delay pipeline facility. An example of radix-2 DIF FFT SFG for N = 16 is depicted in Fig. 1.

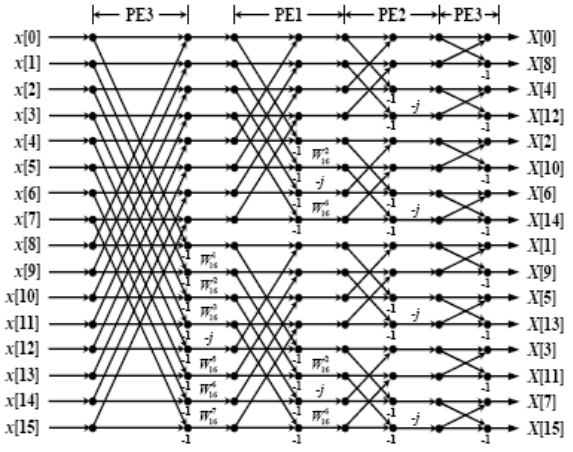

Fig. 1 Radix-2 DIF FFT signal-flow graph of length 8

The radix-2 DIF FFT described above appears regularity in SFG and has less complex multipliers required. Thus, it is suited for hardware implementation, because some complex multiplications can be simplified to reduce the chip area. For instance, an input signal multiplied by W_8^2in Fig. 1 can beexpressed as:**.**

$$(a+jb)W_{16}^2 = \sqrt{2}[(a+b)+j(b-a)]/2, \qquad (2)$$

Where (a+jb) denotes a discrete-time signal in complex form similarly, the complex multiplication of $W_{16}^2$is given by:

$$(a+jb)W_{16}^2 = \sqrt{2}[(b-a)-j(b+a)]/2, \qquad (3)$$

Both these above equations will ease hardware implementation in the future, because they only need to calculate the multiplication by √2 / 2 and two real additions, respectively. Especially, the multiplication by √2 / 2 can be obtained easily, which circuit design will be introduced in the latter section. The inverse discrete Fourier transform (IDFT) of length *N* is given by:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k W_N^{-kn}, \quad 0 \le n \le N-1 \qquad (4)$$

To reuse the same hardware core for reducing the chip area [16], (4) can be rewrite as:

$$x_n = \frac{1}{N}(\sum_{k=0}^{N-1} X_k^* W_N^{-kn})^* \quad 0 \le n \le N-1 \qquad (5)$$

Where the star symbol * denotes a conjugate. This new form can be viewed as a general DFT. In other words, DFT and IDFT can reuse the same hardware core, while IDFT requires some extra computations. These extra computations include conjugating the input data $X_k$and the outcomes of DFT, as well as dividing the previous output by *N*. Obviously, this new reuse version of DFT/IDFT algorithm will also simplify the design effort of an DFT/ IDFT processor and thus reduce the chip area, if both the DFT and IDFT processors are activated alternatively, and not simultaneously.

III. PROPOSED ARCHITECTURE

Traditional hardware implementation of FFT/IFFT processors usually employs a ROM to look up the wanted twiddle factors, and then word length complex multipliers to perform FFT computing. However, this introduces more hardware cost, thus a bit-parallel complex constant multiplication scheme [8] is used to improve the foregoing issue.

Besides, since the twiddle factors have a symmetric property, the complex multiplications used in FFT computation can be one of the following three operation types

$$W_N^k.(a+jb) = W_N^{k-\left(\frac{N}{4}\right)}(b-ja), \quad N/4<k<N/2, \qquad (6)$$

$$W_N^k.(a+jb) = -W_N^{k-\left(\frac{N}{2}\right)}(b-ja), \quad N/2<k<3N/4, \qquad (7)$$

$$W_N^k.(a+jb) = -W_N^{k-\left(\frac{3N}{4}\right)}(b-ja) \quad 3N/4<k<N \qquad (8)$$

Given the above three equations, any twiddle factor can be obtained by a combination of these twiddle-factor primary elements. In other words, arbitrary twiddle factor used in FFT can utilize these operation types to derive the wanted value, thus can significantly shorten the size of ROM used to store the twiddle factors. Moreover, for hardware implementation consideration, we add two extra operation types to further decrease the size of ROM. Our method can also prune away the critical path in the designed hardware such that the system clock becomes faster. The two additional operation types are given by:

$$W_N^k.(a+jb) = [W_N^{(N/4)-k}(b+ja)]^*, \quad 1 \le k < N/4 \qquad (9)$$

$$W_N^k.(a+jb) = -j[W_N^{\left(\frac{N}{2}\right)-k}(b+ja)]^*, \quad N/4 \le k < N/2, \qquad (10)$$

A. Proposed Architecture

In order to improve the previous works on power reduction, we propose a radix-2 pipeline FFT/IFFT processor with low power consumption. The proposed architecture is composed of three different types of processing elements (PEs), a complex constant multiplier, delay-line (DL) buffers (as shown by a rectangle with a number inside), and some extra processing units for computing IFFT. Here, the conjugate for extra processing units is easy to implement, which only takes the 2's complement of the imaginary part of a complex value. In addition, for a complex constant multiplier in Fig. 2, we propose a novel reconfigurable complex constant multiplier to eliminate the twiddle-factor ROM. This new multiplication structure thus becomes the key component in reducing the chip area and power consumption of our proposed FFT processor. The detailed functions of these modules appeared in Fig. 2 are described in the following subsections.

B. Processing Elements

Based on the radix-2 FFT algorithm, the three types of processing elements (PE3, PE2, and PE1) used in our design are illustrated in Fig. 2, Fig. 4, and Fig. 3, respectively. The functions of these three PE types correspond to each of the butterfly stages as shown in Fig. 1. First, the PE3 stage is used to implement a simple radix-2 butterfly structure only, and serves as the sub modules of the PE2 and PE1 stages. In the figure, I_{in} and I_{out} are the real parts of the input and output data, respectively. Q_{in} and Q_{out} denotes the image parts of the input and output data, respectively. Similarly, $DL\text{-}I_{in}$ and $DL\text{-}I_{out}$ stand for the real parts of input and output of the DL buffers, and $DL\text{-}Q_{in}$ and $DL\text{-}Q_{out}$ are for the image parts, respectivelyAs for the PE2 stage, it is required to compute the multiplication by –j or 1. Note that the multiplication by -1 in Fig. 3 is practically to take the 2's complement of its input value.

In the PE1 stage, the calculation is more complex than the PE2 stage, which is responsible for computing the multiplications by –j, $W_N^{N/8}$, and $W_N^{3N/8}$ respectively. Since $W_N^{3N/8} = -j\, W_N^{N/8}$ it can be given by either the multiplication by $W_N^{N/8}$ first and then the multiplication by –j or the reverse of the previous calculation. Hence, the designed hardware utilizes this kind of cascaded calculation and multiplexers to realize all the necessary calculations of the PE1 stage. This manner can also save a bit-parallel multiplier for computing, which further forms a low-cost hardware.

C. Bit-Parallel Multipliers

In Section II, the multiplication by $1/\sqrt{2}$ can employ a bit parallel multiplier to replace the wordlength multiplier and square root evaluation for chip area reduction. The bit-parallel operation in terms of power of 2 is given by:

$$\text{Output} = in \times \sqrt{2}/2 = in \times (2^{-1}+2^{-3}+2^{-4}+2^{-6}+2^{-8}+2^{-14}), \quad (11)$$

If a straightforward implementation for the above equation is adopted, it will introduce a poor precision due to the truncation error and will spend more hardware cost. Therefore, to improve the precision and hardware cost, Eq.(11) can be rewritten as:

$$\text{Output} = in \times \sqrt{2}/2 = in \times [(1+(1+2^{-2})(2^{-6}-2^{-2})], \quad (12)$$

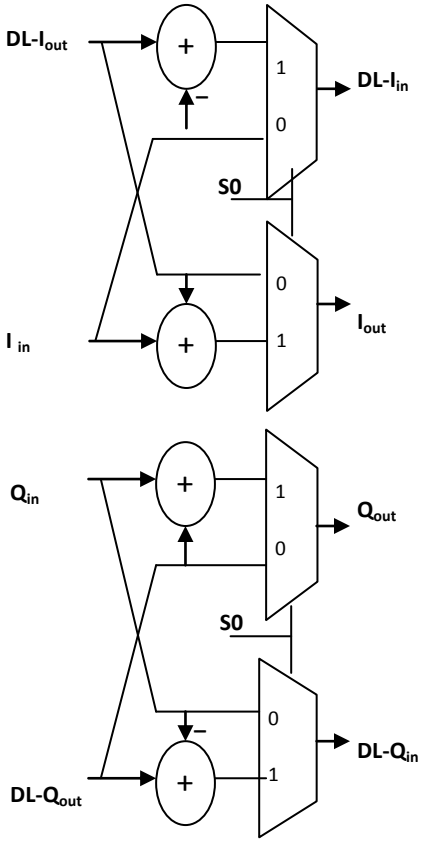

Fig. 2 Circuit diagram of our proposed PE3 stage.

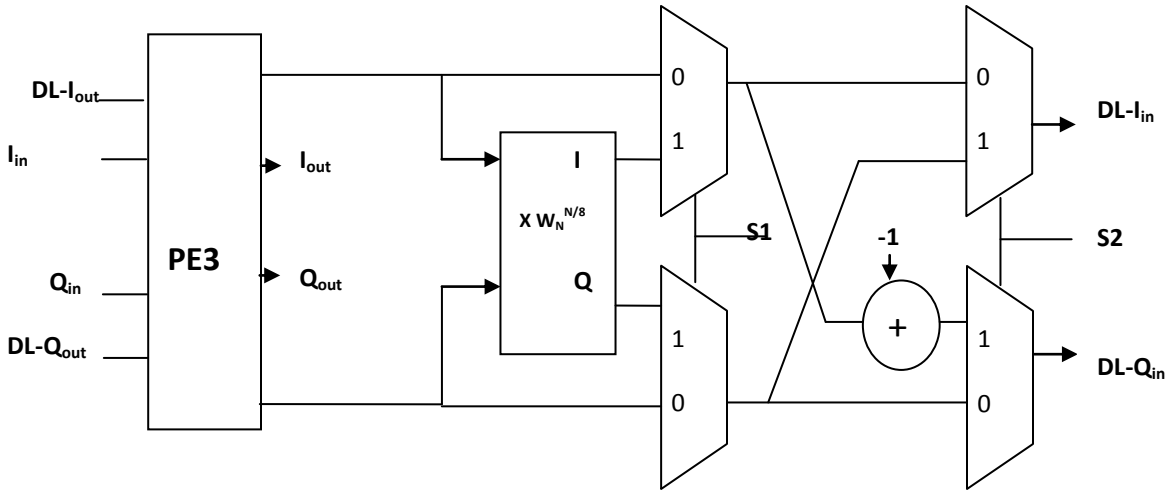

Fig.3 Circuit diagram of our proposed PE1 stage

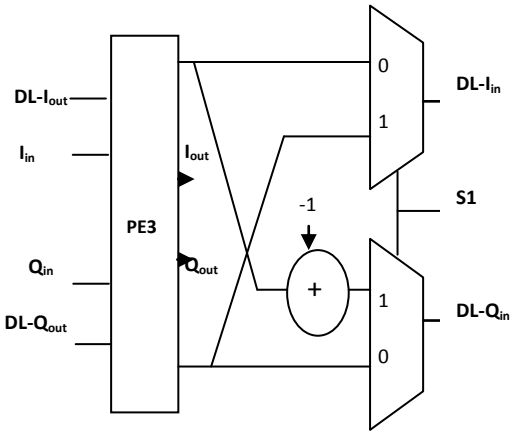

Fig. 4 Circuit diagram of our proposed PE2 stage.

According to , the circuit diagram of the bit-parallel multiplier is illustrated in Fig. 5. The resulting circuit uses three additions and three barrel shift operations. The realization of complex multiplication by $W_N^{N/8}$using a radix-2 butterfly structure with its both outputs commonly multiplied by $1/\sqrt{2}$ is shown in Fig. 6. This circuit has just been used in the PE1 stage.

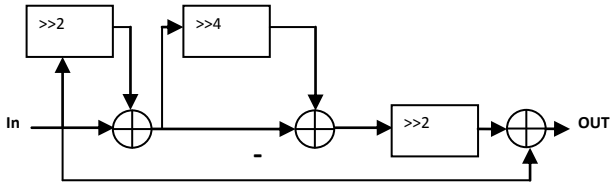

Fig. 5 Circuit diagram of the bit-parallel multiplication by $1/\sqrt{2}$

Besides, we need not to use bit-parallel multipliers to replace the word length one for two reasons. One is on the operation rate. If bit-parallel multipliers are used, the clock rate is decreased due to the many cascades adders. The other reason is the introduction of high wiring complexity because many bit-parallel multipliers are required to be switched for performing multiplication operations with different twiddle factors. In fact, this phenomenon also appears in [8]. Based on the above two reasons, the word of operation speed and chip area. Note that our proposed complex constant multiplier will not length multiplier is still adopted to implement our complex constant multiplier under the consideration. Introduce the issue of high hardware cost as described earlier, because no ROM is used

IV. PERFORMANCE EVALUATION AND RESULT.

The performance evaluation can be obtained by formulation of normalization power per FFT is defined as follows:

Normalized power per FFT =

$$\frac{power}{(voltage)^2(FFT\ sizefrequency)} X\ 1000 \qquad (13)$$

The functional simulation of the proposed architecture has been justified by using Verilog HDL. The result evidences the validation of the proposed architecture. To further validate our proposed architecture, we implement this architecture on a commercial FPGA chip. The result shows that the proposed architecture works very well.

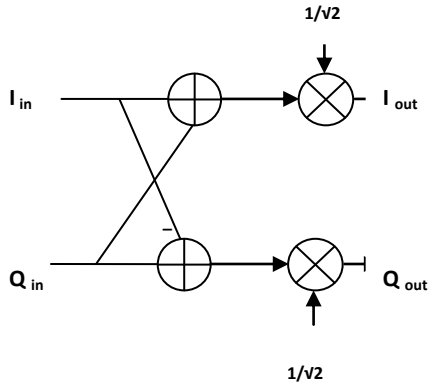

Fig. 6 Circuit diagram of the multiplication by $W_N^{N/8}$

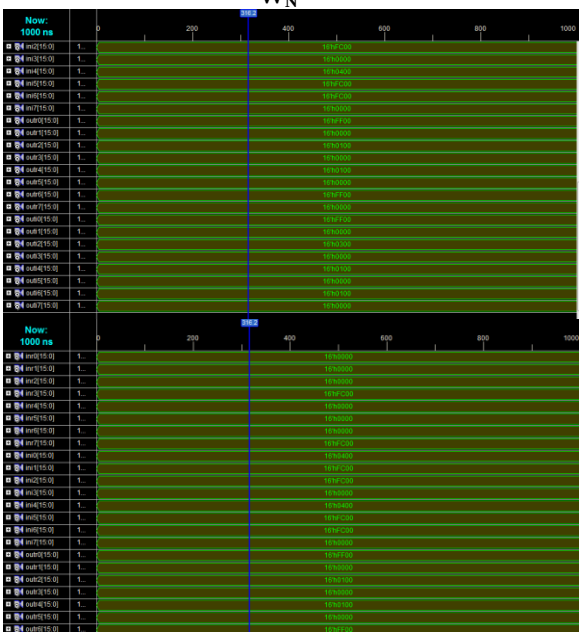

Fig: 7 FFT using proposed architecture

V. CONCLUSION

A novel ROM-less and low-power pipeline FFT/IFFT for OFDM applications have been described in this paper. Considering the symmetric property of twiddle factors in FFT, we have designed a reconfigurable complex constant multiplier such that the size of twiddle factor ROM is significantly shrunk, especially no ROM is needed in our work. This result shows that our design owns lower hardware cost and power consumption compared to the existing ones. Of course, our proposed scheme can also be adapted to high-point FFT applications, with a lower size of twiddle-factor ROM's. our design is relatively low cost and consumes lower power, it can serve as a powerful FFT/IFFT processor in many other wireless communication systems.

REFERENCES

[1] IEEE Std 802.11a, 1999, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-Speed Physical Layer in the 5 GHz band."

[2] IEEE 802.16, IEEE Standard for Air Interface for Fixed Broadband Wireless Access Systems, the Institute of Electrical and Electronics Engineers, Inc., June 2004.

[3] Chu Yu, Yi-Ting Liao, Mao-Hsu Yen, Pao-Ann Hsiung, and Sao-Jie Chen, "A Novel Low-Power 64-point Pipelined FFT/IFFT Processor for OFDM Applications," in *Proc. IEEE Int'l Conference on Consumer Electronics*. Jan. 2011, pp. 452-453.

[4] ETSI, "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television," ETSI EN 300744 v1.4.1, 2001.

[5] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series" *Math. Comput*, vol. 19, pp. 297- 301, Apr. 1965.

[6] S.He and M. Torkelson, "Designing Pipeline FFT Processor for OFDM (de)Modulation," in *Proc. URSI Int. Symp. Signals, Systems, and Electronics*, vol. 29, Oct.1998, pp. 257-262.

[7] H.L. Groginsky and G.A. Works, "A pipeline fast Fourier transform," *IEEE Transactions on Computers*, vol. C-19, no. 11, pp. 1015-1019, Nov. 1970.

[8] KoushikMaharatna, Eckhard Grass, and Ulrich Jagdhold, "A 64-Point Fourier transform chip for high-speed wireless LAN application using OFDM," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 484- 493, Mar. 2004.

[9] Y.T. Lin, P.Y. Tsai and T.D. Chiueh, "Low-power variable-length fast Fourier transform processor," *IEE Proc. Comput. Digit. Tech*., vol. 152, no. 4, pp. 499-506, July 2005.

K.Indirapriyadarsini: studying M.Tech in Swarnandhra College of engineering and technology, Narsapuram, and Major working areas are VLSI and embedded systems Presented research paper in one national conference.

S.kamalakumari: Associate. Professor in swarnandhra college of engineering and technology, Narsapuram, Major working areas are wireless communications, Linear and Digital ICs and VLSI. Has seven years of teaching experience presented research papers in two national conferences.

G.Prasanna Kumar: Asst. Professor in Vishnu institute of technology. Has four years of teaching experience. Major working areas are Digital Signal Processing, Wireless communications and Embedded Systems Presented researchpaper in one national conference.