International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 3, Issue 2 (September 2013) PP: 67-70

Design Radix-4 64-Point Pipeline FFT/IFFT Processor for Wireless Application

Kandhi Srikanth
**P.G Scholar, Sreenidhi Institute of Technology, and Science*

ABSTRACT: *4G and other wireless systems are currently hot topics of research and development in the communication field. Broadband wireless systems based on orthogonal frequency division multiplexing (OFDM) often require an inverse fast Fourier transform (IFFT) to produce multiple sub-carriers. Fast Fourier transform (FFT) processing is one of the key procedures in popular orthogonal frequency division multiplexing (OFDM) communication systems. Our proposed design is aimed to Eliminate the read-only memories (ROM's) used to store the twiddle factors, which requires more memory and also increases chip area and to reduce the number of computations to in order to speed up FFT computation. To remove the read only memories (ROM's) used to store the twiddle factors, our proposed architecture applies various periodicity properties of twiddle factors and reconfigurable multiplier. To reuse the same hardware core for reducing the chip area we implemented single hardware core to perform both FFT and IFFT by adding few extra computations. We implement the processor in SDF architecture with radix-4 algorithm. There are various algorithms to implement FFT, such as radix-2, radix-4 and split-radix with arbitrary sizes .Radix-2 algorithm is the simplest one, but its calculation of addition and multiplication is more than radix-4's. The proposed architecture is compared with the existing architecture for wireless applications based on performance metrics like power consumption, chip area, speed etc.*

Key words: *FFT, IFFT, OFDM, pipeline, Radix-4.*

I. INTRODUCTION

The Fast Fourier Transform (FFT) and its Inverse Fast Fourier Transform (IFFT) are essential in the field of digital signal processing (DSP), widely used in communication systems, especially in orthogonal frequency division multiplexing(OFDM) systems, wireless-LAN, ADSL, VDSL systems and WIMAX . Apart from the applications, the system demands high speed of operation, efficient area and low power consumption,

By considering these facts, we proposed the processor with single path delay feedback (SDF) pipeline architecture. The SDF pipelined architecture is used for the high-throughput in FFT processor. There are three types of pipeline structures; they are single-path delay feedback (SDF), single-path delay commutator (SDC) and multi-path delay commutator. The advantages of single-path delay feedback (SDF) are (1) This SDF architecture is very simple to implement the different length FFT. (2) The required registers in SDF architecture is less than MDC and SDC architectures. (3) The control unit of SDF architecture is easier.

We implement the processor in SDF architecture with radix-4 algorithm. There are various algorithms to implement FFT, such as radix-2, radix-4 and split-radix with arbitrary sizes [3]-[5]. Radix-2 algorithm is the simplest one, but its calculation of addition and multiplication is more than radix-4's. Though being more efficient than radix-2, radix-4 only can process 4n-point FFT. The radix-4 FFT equation essentially combines two stages of a radix-2 FFT into one, so that half as many stages are required. Since the radix-4 FFT requires fewer stages and butterflies than the radix 2 FFT, the computations of FFT can be further improved

In order to speed up the FFT computation we increase the radix, for reducing the chip size we use ROM-less architecture and for further low power consumption we implement the reconfigurable complex multiplier and delay line buffers[8]-[10]. Finally, this paper is organized as follows. Section II describes the FFT/IFFT processor with radix- 4 algorithm. Proposed architecture is discussed in Section III. Finally conclusion and future work will see in Section IV.

II. SECTION II. FFT/IFFT ALGORITHM

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 DIF decomposition because it matches the manipulation manner of single-path delay pipeline facility.

In order to improve the power reduction, we propose a radix-4 64-point pipeline FFT/IFFT processor. The radix-4 FFT algorithm is most popular and has the potential to satisfy the current need. The radix-4 FFT equation essentially combines two stages of a radix-2 FFT into one, so that half as many stages are required. To calculate 16-point FFT, the radix-2 takes $\log_2 16$=4 stages but the radix-4 takes only $\log_4 16$=2stages. A 16-point, radix-4 decimation-in-frequency FFT algorithm is shown in Figure 1. Here, our work employs DIF decomposition because it matches the manipulation manner of single-path delay pipeline facility

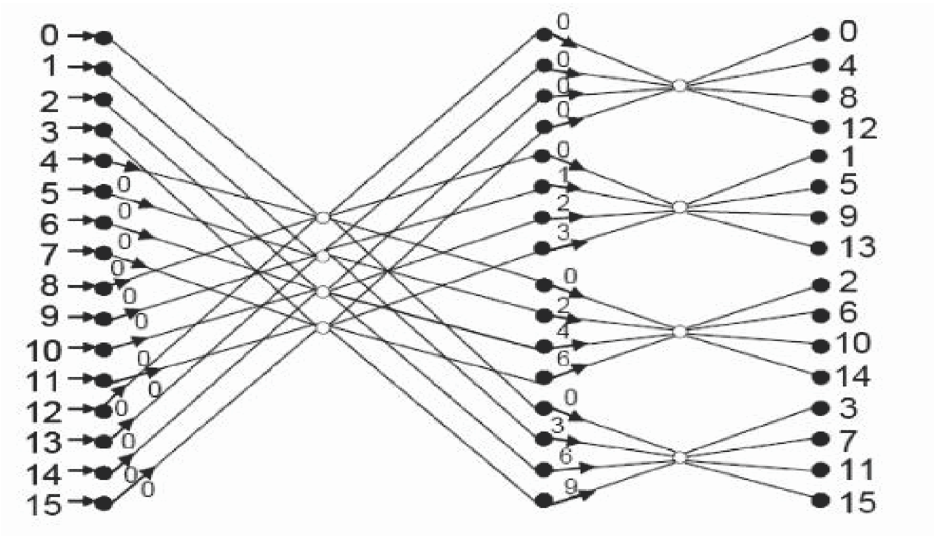

Fig.1: Flow graph of a 16-point radix-4 FFT algorithm

III. PROPOSED ARCHITECTURE

The inverse discrete Fourier transform (IDFT) of length N is given by:

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

To reuse the same hardware core for reducing the chip area above equation can be rewrite as:

$$x_n = \frac{1}{N}\left(\sum_{k=0}^{N-1} X_k^* W_N^{nk}\right)^* \quad , \quad 0 \leq n \leq N-1 \qquad (2)$$

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 Xk 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.

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]-[11], [14]-[18] 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:

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

$$W_N^k.(a+jb) = -W_N^{k-\frac{N}{2}} . (a + j b) \qquad N/2<k<3N/4 \qquad (4)$$

$$W_N^k.(a+jb) = -W_N^{k-\left(3\frac{N}{4}\right)} .(b - j a) \qquad 3N/4<k<N \qquad (5)$$

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. Based on the above operation types above, the 49 complex multiplications after the third butterfly stage for 64-point FFT will be reduced to the computation of 16 primary elements. Clearly, this results in the twiddle-factor ROM table to be shrunk significantly.

Radix-4 Single- path Delay Feedback (Fig. 2) uses the registers more efficiently by storing one output of each butterfly in feedback shift registers. A single data stream goes through the multiplier at every stage. It has same number of butterfly units and multipliers as in R2MDC approach, but with reduced memory requirement (N – 1) registers. Its data throughput is (1/x) times that of the corresponding Rx MDC architecture. The lower area requirement for this structure compared to MDC structures is to its advantage.

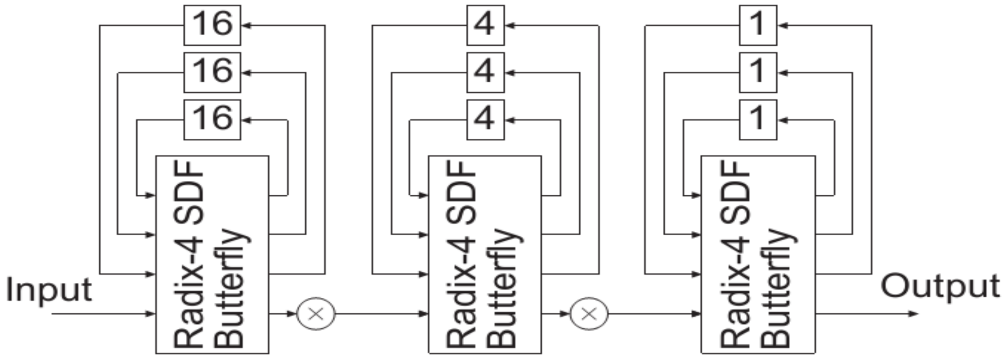

Fig .2 64 point Radix-4 single path delay feedback architecture

Reconfigurable Complex constant Multipliers:

Based on equations (3)-(5), a reconfigurable low-complexity complex constant multiplier for computing W^{i}_{64} is proposed, as shown in Fig. 3.

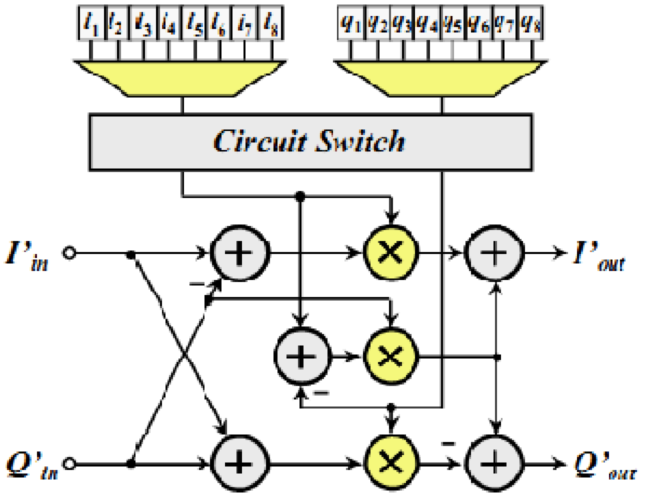

Fig.3 Complex multiplier used in proposed architecture.

This structure of this complex multiplier also adopts a cascaded scheme to achieve low-cost hardware. Here, the meaning of two input signals (I in and I out) and two output signals (Q in and Q out) are the output of butterfly stages.

In many DFT computations, both complex multiplications and real multiplications are required. For the purpose of comparison, the counting is based on the number of real multiplications. A complex multiplication can be realized directly with 4 real multiplications and 2 real additions. With a simple transformation, the number of real multiplications can be reduced to 3, but the number of real additions increases to 3.We considers a complex multiplication as 3 real multiplications and 3 real additions in complex multiplier shown above.

IV. CONCLUSION

A novel ROM-less and low-power pipeline 64-point FFT/IFFT processor for OFDM application has 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. Our design takes 3365 slices when synthesized selecting the device xilinx 3s500e. Of course, our proposed scheme can also be adapted to high-point FFT applications, with a lower size of twiddle-factor ROM’s. So our design is relatively low cost .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] 3GPP LTE, “Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation” 3GPP TS 36.211 v8.5.0, 2008-12. [4] ETSI, “Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television,” ETSI EN 300 744 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] 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] Koushik Maharatna, 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,

[10] 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.

[11] Chin-Teng Lin, Yuan-Chu Yu, and Lan-Da Van, “A low-power 64-point FFT-IFFT design for IEEE 802.11a WLAN application,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2006, pp. 4523-4526.

[12] Yuan Chen, Yu-Wei Lin, and Chen-Yi Lee, “A Block Scaling FFT/IFFT Processor for WiMAX Applications,” in Proc. IEEE Asian Solid-state Circuits Conf., 2006, pp. 203-206.

[13] Sheng-Yeng Peng, Kai-Ting Shr, Chao-Ming Chen, Yuan-Hao Huang, “Energy-Efficient 128rv2048/1536-point FFT Processor with Resource Block Mapping for 3GPP-LTE system,” in Proc. 2010 International Conference on Green Circuits and Systems (ICGCS), 2010, pp. 14-17

[14] Minhyeok Shin and Hanho Lee, “A High-Speed Four-Parallel Radix-24 FFT/IFFT Processor for UWB Applications,” in Proc. IEEE Int. Symp. Circuits and Systems, 2008, pp. 960-963.

[15] Jia Lihong, Gao Yonghong, Isoaho Jouni, and Tenhunen Hannu, “A new VLSI-oriented FFT algorithm and implementation,” in Proc. IEEE International on ASIC Conference, Sept. 1998, pp. 337–341.

[16] Y.-W. Lin,H.- Y.Liu, and C.-Y. Lee, “A 1 GS/s FFT/IFFT processor for UWB applications,” IEEE Journal of Solid-State Circuits, vol. 40, no. 8, pp. 1726-1735, Aug. 2005.