A Low Multiplicative Complexity of Proposed FFT/IFFTs Design Applied for OFDM-Based Wireless Communication Systems

M. Arioua and M. M. Hassani

Abstract—Low arithmetic complexity and High speed of FFT/IFFT processor is required in many applications in OFDM-based wireless broadband communication systems. For this reason, it's essential to develop an optimum complexity design FFT/IFFT processor to meet the low power and real time requirements. This paper presents a multiplicative comparison between developed FFT/IFFTs design and efficient pipeline radix structures applied for OFDM modulator/demodulator to be used in IEEE 802.11a and IEEE 802.16a systems. The proposed design reduces the multiplicative complexity by using feedback architecture and reduction approach of complex multiplications. Based on the algorithm and architecture analysis of developed efficient FFT/IFFTs and comparison results, the proposed designs reaches low arithmetic complexity, hence high speed and low power consumption for OFDM-based wireless broadband communication systems.

Index Terms—FFT/IFFT, FPGA, IEEE 802.11a, IEEE 802.16a, MDC, OFDM, Radix-n, SDF, WLAN.

I. INTRODUCTION

The Fast Fourier transforms (FFT) and its inverse (IFFT) is one of the fundamental algorithms in the field of digital signal processing [1], [2]. The FFT/IFFT is widely used in various areas such as telecommunications, speech and image processing, etc. Recently, this algorithm has been widely used as one of the key components of the Physical layer (PHY) of Orthogonal Frequency Division Multiplexing (OFDM) based wireless broadband communication system [3]. It's one of the highest computational complexity modules of several wireless standards PHY layer (OFDM-802.11a/g, MIMO-OFDM 802.11n, Hyperlan-2, and OFDM-802.16a/d/e) [2].

Due to the increasing demand of OFDM-based applications and high throughput in Wireless Local Area Network (WLAN) and Wireless Metropolitan Area Network (WMAN), the processing speed, power and area implementation are very worthwhile and significant parameter in Fast Fourier Transform design [4]. Because of the complexity of the processing algorithm of FFT/IFFT and its importance in OFDM-based wide band communication systems, the study of algorithms methods and high-performance VLSI

FFT/IFFT architectures is of increasing importance for improving the computation performance [5], [6].

Currently, the main restrictions for FFT/IFFT processors applied for OFDM-based wireless LAN systems are area, power consumption and execution time [4]. N-point FFT/IFFT design in IEEE 802.11a and IEEE 802.16a systems have to fulfill a time and power constraints to perform the computation of FFT/IFFT processor [7], [8]. To meet those criterions, the FFT/IFFT design has to employ a highly in-depth architecture or use a very high operation frequency. One solution will satisfy the timing constraint, but at the cost of other constraints: the area and power consumption [1], [4].

In this paper, we simulated the multiplicative comparison of developed optimum FFT/IFFT design applied for N point processor, which has low arithmetic complexity and high architecture regularity, in the same time satisfies the timing and power consumption of IEEE 802.11a and IEEE 802.16a specifications. This design has been compared to others enhanced FFT/IFFT architectures. The efficient N-points FFT/IFFT design proposed in this paper, based on FPGA platform, gives an advantage in terms of multiplicative complexity, therefore a gain in resource usage and power dissipation using pipelined method and complex multiplication reduction approach.

II. OPTIMUM COMPLEXITY OF THE PROPOSED FFT/IFFT PROCESSOR

A. Architectural Description

The proposed architecture of N-points FFT/IFFT processor internally uses $\log_2 N$ modules of r-points FFT/IFFT for computation. The N-points- Radix-r decimation of the FFT/IFFT can be formulated in the following way:

$$FFT(x) = X[k] = \sum_{n=0}^N x[n] \cdot W_N^{nk} \quad (1)$$

We suppose: $k=i+r.t$ and $n=j+r.m$
where $i, j \in \{0, 1, \dots, r-1\}$ and $m, t \in \{0, 1, \dots, r-1\}$.

The N-points FFT equation in (1) can be expressed as:

$$X[i+r.t] = \sum_{j,m=0}^N x[j+r.m] \cdot W_N^{(j+r.m)(i+r.t)} \quad (2)$$

After simplifying, we obtain the equation (3):

$$X[i+rt] = \sum_{j=0}^{r-1} [W_N^{ij} \sum_{m=0}^{r-1} x[j+rm] \cdot W_r^{im}] \cdot W_4^{jt} \quad (3)$$

Manuscript received August 10, 2013; revised November 5, 2013.

M. Arioua is with the National School of Applied Sciences, University of Abdelmalek Essaadi Tetouan-Tanger, B.P.2222 M'hannech, Morocco (email: m.arioua@ieee.org).

M. M. Hassani is with the Faculty of Science of Marrakech at Cadi Ayyad University, B.P.511- 40000, Morocco (email: hassani@ucam.ac.ma).

As demonstrated in formula (3), the implementation of the FFT algorithm for computation of the N -points FFT (i.e $N=r^2$) involves computation of two r -points FFT. This one can be decimated by using a Radix- r algorithm (Fig. 1).

Fig. 1. N -points FFT computation using Radix- r algorithm.

Fig. 2. N -points FFT/IFFT proposed design.

The first r -points module computes r -points of the N -points FFT on the fitting data slot according to equation (3) and then multiplies the output with $r \times r$ inter-dimensional constants coefficients by a multiplier and once again computing the r -points FFT of the resultant data with the fitting data reordering according to equation (3).

With this proposed design we implemented $N=16$ -points FFT/IFFT [9], $N=64$ -points FFT/IFFT [10] and $N=256$ -points FFT/IFFT. For instance, the $N=64$ -points FFT/IFFT equation (2) can be expressed as:

$$X[i + 8t] = \sum_{j=0}^7 [W_{64}^{ij} \sum_{m=0}^7 x(j + 8.m) W_8^{im}] W_8^{jt} \quad (4)$$

This one can be decimated by using Radix2, Radix-4, or Radix-8. The Radix-8 algorithm is an attractive algorithm for its requirement of less complex multiplications and additions/subtractions comparing to Radix-2 and Radix-4 algorithms, however, the use of algorithms with high radix degree increase the complexity of integration in an integrated circuit [11], [12].

Even if the number of non-trivial additions and multiplications present a good clue on the effectiveness of an algorithm, hardware integration considers algorithm regularity as well as the conception complexity and architecture control. The gains achieved by the reduction of multiplications or additions could be sometimes lost by the control complexity induced and the interconnection surplus. Hence the interest goes to Radix-2 algorithm that offers more large regularity for architectural hardware implementation compared to Radix-8 and Radix-4 and Split-Radix.

The Radix-2 algorithm is appealing for its simplicity but has the disadvantage for being not adapted for very large points FFT/IFFT calculation, due to the high multiplier requirement. However, to reduce the power consumption in the N -point FFT/IFFT, the number of complex multiplications must be reduced. For this reason, we applied the proposed design methodology for efficient N -points FFT processor based on Radix-2 r -points FFT module in order to keep the area and power consumption as low as possible, meanwhile respecting the restriction of the execution time (Fig. 2).

The r -points FFT module uses Radix-2 algorithms for computation. After the computation of the initial input data sequences in the r -points FFT, the de-multiplexer holds the data to a serial parallel bloc in the first FFT/IFFT computation or to N -points FFT output in the second computation. In the first case the data are arranged for multiplication in the multiplier unit according to equation (3).

The arranged data undergoes an operation of inter-dimensional complex multiplication. The multiplier unit should normally perform complex multiplications with N elements. However, in practice a complex multiplication reduction approach leads to a significant reduction of complex multiplications at the cost of far less expensive additions. Indeed, adders require less hardware in FPGA platform and consume much less power than multipliers for the same word-length [13]. Moreover, they have fewer glitches. The implementation complexity of non-trivial twiddle factors reduced even further, due to replacement of the complex multiplications by basic operations, such as shift and adds operations [14].

B. Multiplicative Comparison

The complex multiplication is an expensive operation in the FFT/IFFT computation. It is the prominent factor that has an effect on the chip area, power consumption, and the throughput of an FFT/IFFT processor intended for an FPGA implementation [11], [13]. Therefore, with a large number of complex multiplications, direct computation will requires a large chip area and high power consumption. An additional gain in terms of real multiplications can be obtained if we consider the reduction method of complex multiplication [15]. We tend to reduce the multiplicative complexity of the twiddle factor inside the butterfly processor by calculating only three real multiplications and three additions/subtractions operations as demonstrated in equation (5), (6) and (7):

The multiplication with complex twiddle factor:

$$R + jI = (X + jY)W_N = (X + jY)(C + jS) \quad (5)$$

However this multiplication can be simplified:

$$R = (C - S)Y + Z \quad (6)$$

$$I = (C + S)X - Z \quad (7)$$

and

$$Z = C(X - Y) \quad (8)$$

In the implementation of the complex multiplication module, the twiddle factors coefficients are known in advance.

i.e, C and S in Equations 6 and 7 are pre-computed and stored in a memory table [16]. In this case, it is necessary to store the following three coefficients C, C+S, and C-S. those constants can be saved as canonical signed digits (CSD) to implement complex multiplication with carry and save tree [17]. Thus, the area and power consumption of the complex multiplier can both be reduced.

The storage operation is used to simplify the complex multiplication, for instance, the complex multiplication with $W_N^k = e^{j\frac{2\pi}{N}k}$ factor in N -points FFT/IFFT computation requires only two real multiplications rather than three multiplications. Moreover, the complex multiplication can be reduced further with an efficient number representation of fixed-point arithmetic [18]. The implemented method of complex multiplication used in this work uses three multiplications, one addition and two subtractions. This is done at the cost of an additional memory table. In the hardware description language (VHDL) program, the twiddle factor multiplier was performed using component instantiations of three lpm-mult and three lpm-add-sub modules from Altera library. Worth to note that lpm modules are supported by most of EDA vendors and LPM provides an architecture-independent library of modules that are parameterized to achieve scalability and adaptability [19].

This method allows us to reduce the number of multiplications of FFT/IFFT processor in a dedicated circuit (FPGA), particularly for large point computation (FFT/IFFT 256-points, 1024-points ...). Therefore, we gain enormously in arithmetic complexity, and afterwards, a significant reduction in area and power consumption of FFT/IFFT processor.

The proposed designs of N -points (16-points, 64-points, and 256-points) FFT/IFFTs were coded in VHDL using Altera software "Quartus" and then simulated and synthesized for Altera Cyclone 2 EP2C35F672C6 device. The purpose is to determine the resource usage of the proposed design. The functional and timing synthesis and simulation were performed.

The proposed designs allows three resource reductions; the complex multiplication reduction inside the multiplier unit, combined with the fact of using pipeline architecture. In addition, this structure has eliminated some complex multiplications inside the r -point FFT/IFFT unit.

The proposed designs requires less number of arithmetic operations in terms of real multiplications compared to the conventional Cooley-Tukey algorithm, and to efficient processor implemented in pipeline architecture for Radix-2, Radix-4, Radix-8 and Split-Radix algorithms. A comparison of the proposed design in the case of 16-points and 64-points FFT/IFFT with different FFT/IFFT pipeline architectures are presented in Table I and Table II. Other multiplicative comparison is proposed for developed 256 points FFT/IFFT. We compared the performance, in terms of the numbers of multiplications, of the three designs with efficient Radix pipeline structures with the same word length. Fig. 3 shows the simulated results.

Applying permutations, shift-and-add operations with twiddle factors inside the N -points FFT module reduces more the number of multiplication operations. Multiplications with

non-trivial twiddle factors W_{16}^n , W_{64}^n , W_{256}^n were implemented with embedded multiplier 9-bit. The employed logic operations allow us to cut down the number of complex multiplications in the proposed approach (case of 16-points FFT: Proposed processor 2). Therefore, the number of real multiplications was reduced (Table I).

TABLE I: COMPARISON OF THE PROPOSED DESIGN OF 16-POINTS FFT/IFFT WITH DIFFERENT FFT/IFFT PIPELINE ARCHITECTURES.

Architectures	Real Multiplications	Real Add/Sub	Embedded Multiplier 9-bit
Cooley-Tukey	128	192	128
Radix-2MDC	24	152	6
Radix-4MDC	20	148	9
Split-Radix MDC	20	148	9
Radix-4SDF	24	148	3
Radix-2SDF	24	152	6
Radix-4SDF	20	148	3
Radix-2 ² SDF	24	152	3
Split-Radix SDF	20	148	3
Proposed processor 1 [9]	16	154	6
Proposed processor 2	12	162	6

Fig. 3. Number of real multiplications in the proposed design and efficient pipeline radix structures.

TABLE II: COMPARISON OF THE PROPOSED DESIGN OF 64-POINTS FFT/IFFT WITH DIFFERENT FFT/IFFT PIPELINE ARCHITECTURES [10].

Architectures	Real Multiplications	Real Add/Sub	Embedded Multiplier 9-bit
Cooley-Tukey	192	1152	192
Radix-2MDC	264	1032	15
Radix-4MDC	208	976	18
Radix-8MDC	204	972	21
Split-Radix MDC	196	964	24
Radix-2SDF	264	1032	15
Radix-4SDF	208	976	6
Radix-2 ² SDF	208	976	6
Split-Radix SDF	196	964	6
Proposed processor	132	1080	21

The N -points FFT/IFFT proposed processors requires less number of arithmetic operations compared to the efficient processors implemented in pipeline structure with Radix-2, Radix-2², Radix-4 and Split-Radix algorithm implemented with MDC and SDF architectures. However, the proposed processor needs more resource usage in terms of embedded multiplier in order to attain low multiplicative complexity (Table I, Table II). Hence, achieve high speed and low power consumption at the expense of losing logic area.

The simulation result shows at Quartus tool that the proposed designs significantly reduce the number of operations within the processor. The proposed modules can be integrated with other components to be used as standalone processor (1024, 2048-points FFT/IFFT) applied for OFDM based Wireless Broadband Communication.

In order to verify the accuracy of computation of the implemented FFT/IFFT cores, we had simulated the calculation FFT/IFFTs in functional simulation mode. The output of the implemented FFT/IFFTs bloc approximately matches the output of FFT/IFFTs function written in Matlab representing a theoretical example of FFT/IFFT calculation.

III. CONCLUSION

In this paper, we have presented an efficient design to realize N -point FFT/IFFTs processors for OFDM-based wireless communication systems (IEEE 802.11a, IEEE 802.16a). The number of multiplications has been used in this work as a key metric for comparing FFT/IFFT structures since it has a large impact on the resource usage and power consumption. The simulation result shows that proposed design significantly reduces the number of multiplication operations inside developed FFT/IFFT processors compared to other efficient processors.

REFERENCES

- [1] B. Zhou, Y. Peng, and D. Hwang, "Pipeline FFT architectures optimized for FPGAs," *International Journal of Reconfigurable Computing*, vol. 2009 (2009), Article ID 219140.
- [2] W. Li, "Studies on implementation of lower power FFT processors, Linköping Studies in Science and Technology," Thesis No. 1030, ISBN 91-7373-692-9, Linköping, Sweden, 2003.
- [3] Y. Jung, H. Yoon, and J. Kim, "New efficient FFT algorithm and pipeline implementation results for OFDM/DMT applications," *IEEE Trans. on Consumer Electronics*, vol. 49, issue 1, 2003.
- [4] A. Saeed, M. Elbably, G. Abdelfadeel, and M. I. Eladwy, "Efficient FPGA implementation of FFT/IFFT processor," *Intern. Journal of Circuit, Systems and Signal Processing*, vol. 3, issue. 3, pp.103-110, 2009.
- [5] N. Weste and D. J. Skeltem, "VLSI for OFDM," *IEEE Communications Magazines*, vol. 36, issue. 10, pp. 127-131, Oct. 1998.
- [6] M. Jaber and D. Massicotte, "A new FFT concept for efficient VLSI Implementation: Part II-parallel pipelined processing," in *Proc. IEEE Int. Conf. Digital Signal Processing*, pp.1-6, 2009.
- [7] M. Merlyn, "FPGA implementation of FFT processor with OFDM transceiver," in *Proc. International Conference on Signal and Image Processing (ICSIP)*, pp. 483-489, Dec. 2010.
- [8] D. Zhi, Z. Yimeng, H. Zhiping, T. Guilin, and L. Chunwu, "Simulation and application of FFT based pipelined stream," in *Proc. International*

Conference on Information Engineering and Computer Science (ICECS), Wuhan, China, pp. 1-4, 2009.

- [9] M. Arioua, S. Belkouch, and M. M. Hassani, "Efficient 16-points FFT/IFFT architecture for OFDM based Wireless Broadband Communication," *Information Technology Journal*, vol. 11, issue 1, pp. 118-125, 2012.
- [10] M. Arioua, S. Belkouch, M.M. Hassani, "Efficient 64-points FFT/IFFT processor based on 8-point FFT Module for OFDM Based WLAN," *International Review on Computers and Software*, vol. 7, issue 1, pp. 92-99, 2012.
- [11] Ch. Sahnine, "Architecture of reconfigurable integrated circuit, very high speed and low consumption for digital processing of the advanced OFDM," Ph.D. Thesis, Polytechnic Institute of Grenoble, France, 2009.
- [12] M. A. Jaber, D. Massicotte, and Y. Achouri, "A higher radix FFT FPGA implementation suitable for OFDM systems," in *Proc. The 18th IEEE International Conference on Electronics Circuits and Systems (ICECS)*, pp. 744-747, Dec. 2011.
- [13] B. Wang, Q. Zhang, T. Ao, and M. Huang, "Design of pipelined FFT processor based on FPGA," in *Proc. of 2nd Intern. Conf. on Computer Modeling and Simulation (ICCMS)*, Hainan, pp. 432-435, 2010.
- [14] M. Petrov, M. Glesner, "Optimal FFT Architecture Selection for OFDM Receivers on FPGA," in *Proc. of IEEE Intern. Conf. on Field Programmable Technology*, pp. 313-314, Singapore, 2005.
- [15] T. Widhe, "Efficient implementation of FFT processing elements," Linköping Studies in Science and Technology Thesis no. 619, Linköping University, Sweden, 1997.
- [16] M. Arioua, S. Belkouch, M. M. Hassani, and M. Agdad, "Complex multiplication reduction in pipeline FFT architecture," in *Proc. 20th International Conference on Computer Theory and Applications (ICCTA)*, Alexandria, Egypt. 2010.
- [17] M. Kannan, and S.K. Srivasta, "Low power hardware implementation of high speed FFT core," *Journal of Computer Science*, vol. 3, issue 6, pp. 376-382, ISSN 1549-3636, 2007.
- [18] S. Simard, J. G. Mailloux, and R. Beguenane, "Optimized FPGA mapping of a bit-serial square root operator with minimum output delay," *International Review on Computers and Software*, vol. 2, issue 6, pp. 661-665, 2007.
- [19] U. M. Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 3rd Ed. Springer, 2007.

M. Arioua received the B.Sc. degree in Telecommunications Engineering and M.S. degree in Telecommunications and Networks from College of Sciences and Technologies of Marrakech in 2005 and 2007, respectively. In 2009, he received M.Sc degree in Multimedia Information Systems, Mobiles and Networks from Picardie Jules Vernes University, Amiens, France. He received his PhD degree in telecommunications and computer

science from Cadi Ayyad University, Marrakech in 2012. Currently, he is professor at National School of Applied Sciences in the department of Engineering Technologies: Telecommunications and Mechatronics, University of Abdelmalek Essaadi Tetouan-Tanger. His research interests include multimedia information systems, networking and communication, wireless communications and mobile computing, real-time processing and embedded systems.

M. M. Hassani received a doctorate thesis in automatic from Nice University, France, in 1982 and Ph. D degree in electrical engineering from Sherbrook University, Canada, in 1992. He is now a professor in the Faculty of Sciences Semlalia, Cadi Ayyad University, Marrakech, Morocco. He heads at the University both the Doctoral Studies Centre of Science and Technology and Electronics and Instrumentation Lab. His research interests are in statistical signal processing, nonlinear system, identification and equalization fields.