Survey of Optimization of FFT Processor for OFDM Receivers

Nirav J. Chauhan[1]
PG Student, E.C. Department, KIT&RC, Kalol, Gujarat, India
niravjchauhan.ec@gmail.com

Shruti Oza[2]
Associate Professor, E.C. department KIT&RC, Kalol, Gujarat, India
swajan_2004@ yahoo.co.in

K.R.Parmar[3]
Associate Professor, L.D. College of Engineering, Ahmedabad, Gujarat, India

Abstract

We will develop FFT processor for some particular OFDM application. In order to achieve this goal, several steps need to be followed. The first step is to find the Specifications for this FFT processor, which is determined on the basis of the application for which the FFT processor is to be designed. After defining the specifications, optimized FFT algorithm and architecture should be used for these specifications. There are a large number of FFT algorithms and architectures in the signal processing literature. Therefore, the state of art algorithms and architectures should be analyzed and compared. Based on different algorithms and architectures, different power consumptions, area and speed of the processor will be achieved. So their ASIC suitability should be analyzed and the effort should be focused on the choosing algorithms and architectures and optimization. Furthermore, the improvement space should be analyzed and the architecture should be further optimized.

The proposed algorithm and architecture should be validated by MATLAB simulation before implementation. After that, it is implemented with VHDL. The synthesis results will be compared with other published FFT processor results.

Keywords: FFT, IFFT, DFT, Pipelining, Parallel Processing, Orthogonal Frequency Division Multiplexing (OFDM).

1. INTRODUCTION

Orthogonal Frequency Division Multiplexing (OFDM) system is famous for its robustness against frequency selective fading channel. The Fast Fourier Transform (FFT) and Inverse FFT (IFFT) processor are used as the modulation/demodulation kernel in the OFDM systems. The sizes of FFT/IFFT Processors are varied in the different applications of OFDM systems.

The terms Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) are used to denote efficient and fast algorithms to compute the Discrete Fourier Transform (DFT) and the Inverse Discrete Fourier Transform (IDFT) respectively. The FFT/IFFT is widely used in many digital signal processing applications and the efficient implementation of the FFT/IFFT is a topic of continuous research.

During the last years, communication systems based on Orthogonal Frequency Division Multiplexing (OFDM) have been an important driver for the research in FFT/IFFT algorithms and their implementation. OFDM is a bandwidth efficient multiple access scheme for digital communications. Many of nowadays most important wireless communication systems use this OFDM technique: Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB) (ETS, 2004), Wireless Local Area Network (WLAN) (IEE, 1999), Wireless Metropolitan Area Network (WMAN) (IEE, 2003) and Multi Band –OFDM Ultra Wide Band (MB–OFDM UWB) (ECM, 2005). Moreover, this technique is also employed in important wired applications such as Asymmetric Digital Subscriber Line (ADSL) or Power Line Communication (PLC).

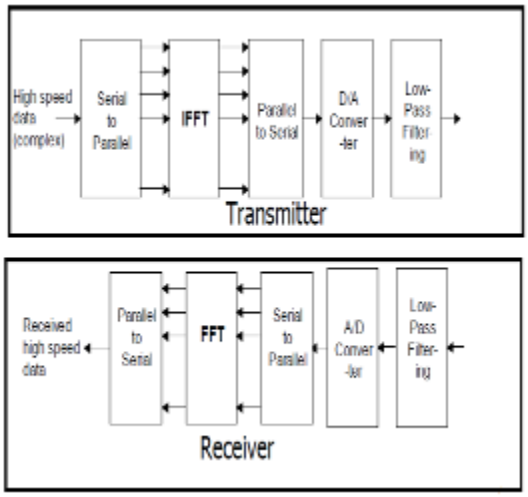

Figure 1.1: Baseband OFDM system

OFDM systems rely on the IFFT for an efficient implementation of the signal modulation on the transmitter side, whereas the FFT is used for efficient demodulation of the received signal. The FFT/IFFT becomes one of the most critical modules in OFDM transceivers. In fact, the most computationally intensive parts of an OFDM system are the IFFT in the transmitter and the Viterbi decoder in the receiver. The FFT is the second computationally intensive part in the receiver. Therefore, the implementation of the FFT and IFFT must be optimized to achieve the required throughput with the minimum penalty in area and power consumption.

2. FFT PROCESSORS IN OFDM RECEIVERS

Different hardware architectures have been used in the literature for the implementation of the CT algorithms. The FFT hardware architectures can be classified into three groups:

• Monoprocessor: A single hardware element is used to perform all the butterflies, twiddle factor multiplications and data shuffling of each stage. The same hardware is reused for all the stages.

• Parallel: The computation of the butterflies, twiddle factor multiplications and data shuffling within one stage is accelerated by using several processing elements. The same hardware elements are again reused for all the stages.

• Pipeline: A single hardware element is used to perform all the butterflies, twiddle factor multiplications and data shuffling of each stage. However, in contrast to former categories, a different hardware element is used to process each stage.

Cooley and Tukey developed the well-known radix-2 Fast Fourier Transform (FFT) algorithm to reduce the computational load of the DFT. The Discrete Fourier Transform (DFT) X(k) of N points is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_{Nnk} \{ 0 \le k < N-1 \}, \quad W_{Nnk} = \exp(-j2\pi nk/N) \ldots (1)$$

Where the X(k) and x(n) are frequency-domain sequences and time-domain sequence.

2.1. Optimization Techniques

The Fast Fourier Transform (FFT) is an efficient algorithm for computing the Discrete Fourier Transform (DFT) and requires less number of computations than that of direct evaluation of DFT. It has several applications in signal processing. Because of the complexity of the processing algorithm of FFT, recently various FFT algorithms have been proposed to meet real-time processing requirements and to reduce hardware complexity over the last decades [1].
There are two directions. One related to the algorithmic point of view and the other based on ASIC architecture. The last one was pushed by VLSI technology evolution. Here, 1024 point FFT processors are designed using three different architectures:

(I) In one of the architecture Twiddle factors are generated using CORDIC (Coordinate Rotation Digital Compute) algorithm.
(II) In another one through Sine/Cosine Look up table it is generated.
(III) Xilinx Logicore FFT processor is also used as useful architecture.
All are designed in FPGA through VHDL [1].

OFDM systems rely on the IFFT for an efficient implementation of the signal modulation on the transmitter side, whereas the FFT is used for efficient demodulation of the received signal. The FFT/IFFT becomes one of the most critical modules in OFDM transceivers.
In fact, the most computationally intensive parts of an OFDM system are the IFFT in the transmitter and the Viterbi decoder in the receiver [6].

The FFT is the second computationally intensive part in the receiver. Therefore, the implementation of the FFT and IFFT must be optimized to achieve the required throughput with the minimum penalty in area and power consumption. The demanding requirements of modern OFDM transceivers lead, in many cases, to the implementation of special–purpose hardware for the most critical parts of the transceiver. Thus, it is common to find the FFT/IFFT implemented as a Very Large Scale Integrated (VLSI) circuit. The techniques applied

to the FFT can be applied to the IFFT as well. Moreover, the IFFT can be easily obtained by manipulating the output of a FFT processor [6].

Many different variations of CT algorithms have been proposed in the literature to improve different aspects of the implementation (memory resources, number of arithmetic operations, etc.) and their mapping to a specific hardware architecture.

Pipeline architectures are well suited to achieve **Small silicon area, High throughput, Short processing time and Reduced power consumption**. Table 3.2 presents the hardware complexity of the candidate pipeline-SDF architectures.

Architec.	r	Normal mults.	Constant mults.	ROM (regs.)	RAM (regs.)	adds /subs
pipeline-SDF	2^2	8	0	80	192	28
pipeline-SDF	2^3	4	4	64	192	40
pipeline-SDF	2^4	4	4	64	192	35

Table 2.1: Hardware complexity of candidate FFT architectures working at60 MHz for WLAN systems [6]

Table 2.2 summarizes the features of some FFT/IFFT processors for OFDM systems proposed in the literature.

FFT points	Architecture	Algorithm	Application
64	Pipeline–MDC	r–2 DIT	WLAN
64	Pipeline	r–2 DIT	WLAN
64	Monoprocessor	r–2 DIT	WLAN
128	Pipeline–MRMDF	MR $2/2^3$ DIF	UWB
128	Pipeline–BRMDC	r–2 DIF	UWB
128	Pipeline–SDF	r–2^4 DIF	UWB
64/128	Pipeline 8-path DF	MR DIF	UWB
128	Pipeline	r–2^4 DIF	UWB
8192	Pipeline–SDC	r–4/2 DIF	DVB–T
8192	Parallel	r–2^3 DIT	DVB–T
2/8 K	Pipeline–SDF	r–4/2 DIF	DVB–T
2/4/8 K	Pipeline–SDF	r–2^2 DIF	DVB–T/H
8 K	Pipeline–SDF	BD	DVB–T
1024	Pipeline–SDF	r–2^2 DIF	OFDM
64–2048	Cached memory	r–2 DIT	OFDM
1024	Monoprocessor	r–4 DIF	OFDM
64	Parallel	r–2 DIT	OFDM
64	Pipeline	r–2 DIF	MIMO
64	Pipeline	r–2 DIT	OFDM
64	Pipeline	r–2 DIF	OFDM
64	Pipeline	MR	OFDM
32 K	Pipeline	r–2^k DIF	DVB–T2

Table 2.2: FFT proposals for OFDM systems [6]

The CORDIC(coordinate rotation digital computer) algorithm can be used to calculate elementary functions with iterative technique, the basic idea is to use a serial of basically deflection angle approaching the desired rotation angle, realizing by using simple hardware such as shifters, adders, sub tractors and comparators without the need of complex multipliers [2].

Based on the analysis, the hardware design of the CORDIC Plural-multiplier is proposed. The complex triangle can be decomposed for unity operation of shift and add/subtract in the CORDIC algorithm, greatly reduced the complexity of the design, it has been widely used in FFT design. This paper has put forward the design of the CORDIC circuit, optimizing mainly from the following three aspects:
(1) Expanding the scope on the whole coordinate system by making pre-treatment of the input angle.
(2) The system gain optimized.
(3) Improve the precision of the operation through increasing input data wide.

The comparison data proves that such a pipelined CORDIC multiplier can be applied to the requirement of fields such as FFT processor [2].

As the Booth multiplier is not suitable for hardware implementations of large FFT, in the paper we propose the CORDIC-based multiplier. Moreover, we develop a ROM-free twiddle factor generator using simple shifters and adders only, 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 [4].

Traditionally, we need lo design various points of FFT/IFFT processors for every application of OFDM system individually, as shown in Table 3.3 This causes the waste of time and money [5].

Application	FFT/IFFT Size	Frequency spacing	T_{FFT}
WLAN	64	0.3125 MHz	3.2 μs
ADSL	2x256	4.3125 KHz	231 μs
VDSL	$2x256x2^n$,n=0:4	4.3125 KHz	231 μs
DAB	$256x2^n$,n=0:3	$4.065x2^n$ KHz	$31x2^n$ μs
DVB-T	8192/2048	,1.116/4.464 KHz	896/224 μs

Table 2.3: FFT/IFFT Size for OFDM Applications

The CORDIC-based 128/256/512/1024/2048/4096/8192-point FFT processor has been implemented by 0.18 µm CMOS with core size of 1897 × 1897 µm2, which takes 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 [4].

3. MATLAB RESULTS

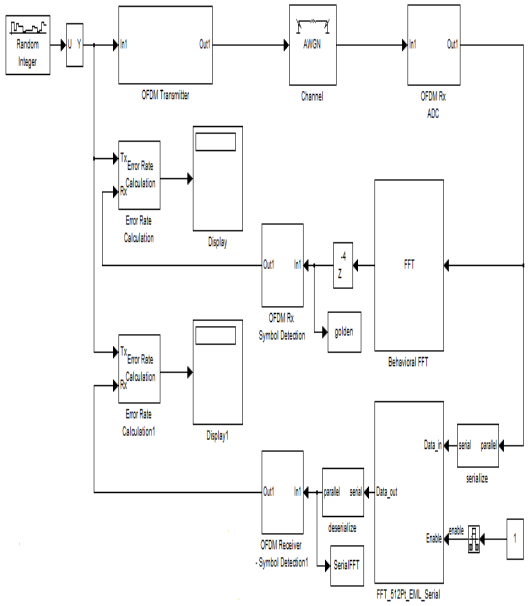

Figure 3.1: General Block-diagram of OFDM receiver with 512-point conventional Cooley-Tukey FFT algorithm

In the figure 3.1 General Block-diagram of OFDM receiver with 512-point conventional Cooley-Tukey FFT algorithm is shown. After implementing it in MATLAB Simulink and adjusting standard parameters the resulting error-rates, number of errors detected and the total number of symbols compared are being displayed in figure 3.2.

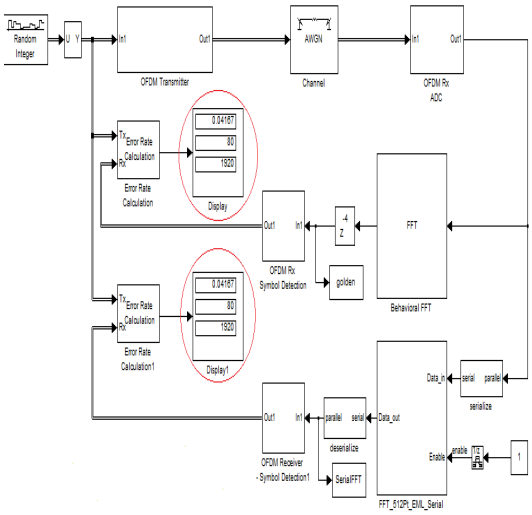

Figure 3.2: General Block-diagram of OFDM receiver with 512-point conventional Cooley-Tukey FFT algorithm

4. CONCLUSION

From different algorithms of FFT processors available in literature it can be said that there is no unique FFT algorithm, architecture and implementation that is optimal for all OFDM systems. Therefore, it is recommended to perform a search across the algorithm, architecture and implementation dimensions for each OFDM system. Different applications of OFDM such as WLAN, DAB, DVB-T, ADSL, VDSL etc. needs different size of FFT processor for modulation/demodulation purpose. Due to Small silicon area, High throughput, Short processing time and Reduced power consumption pipeline architecture is well suited and It is also possible to develop CORDIC based ROM-free twiddle factor generator using simple shifters and adders only, which obviates the need to store all the twiddle factors in a large ROM space hence great reduction in complexity of design.

Here, resulting error-rates, number of errors detected and the total number of symbols for conventional FFT has been successfully simulated and the FFT Processor's parameters can be varied and then optimized to get better results.

REFERENCES

[1] Debalina Ghosh, Depanwita Debnath, Dr. Amlan Chakrabarti, "FPGA Based Implementation of FFT Processor Using Different Architectures", IJAITI - 2012.

[2] Shi Jiangyi, Wang Mingxing , Tian Yinghui, Yang Zhe, "Design of Plural-Multiplier Based on CORDIC Algorithm for FFT Application", ISBN - 978-0-7695-4608-7/12, Second International Conference on Intelligent System Design and Engineering Application – 2012.

[3] Jing He, Jun Wang, Xinyu Xu, "Word-length Optimization of A Pipelined FFT Processor ", ISBN - 978-1-61284-459-6/11, IEEE – 2011.

[4] Tze-Yun Sung, Hsi-Chin Hsin, Yi-Peng Cheng, "Low-power and high-speed CORDIC-based split-radix FFT processor for OFDM systems", Elsevier Inc., pp – 511-527, 2010.

[5] Jen-Chih Kuo, Ching-Hua Wen and An-Yeu (Andy) Wu, "Implementation of a Programmable 64-2048 point FFT/IFFT Processor", ISBN - 0-7803-776 I-3/03, IEEE – 2003.

[6] A. Cortes, I. Velez, M. Turrillas and J. F. Sevillano (2012). Fast Fourier Transform Processors: Implementing FFT and IFFT Cores for OFDM Communication Systems, Fourier Transform - Signal Processing, Dr Salih Salih (Ed.), ISBN: 978-953-51-0453-7, InTech, Available from: http://www.intechopen.com/books/fouriertransform - signal-processing/fast-fourier-transform-processors-implementing-fft-and-ifft-cores-for-ofdm-communication-systems.

[7] MATLAB Toolbox.

[8] Maria Jose Canet, Javier Valls, Vicenc Almenar, Jose Marin-Roig, "FPGA implementation of an OFDM-based WLAN receiver", Elsevier Inc., pp – 232-244, 2012.

[9] O. Font-Bach, N. Bartzoudis, A. Pascual-Iserte, D. Lopez Bueno, "A real-time MIMO-OFDM mobile WiMAX receiver: Architecture, design and FPGA implementation", Elsevier Inc., pp – 3634-3647, 2011.