

# A Study of Optimal Bit Truncation in FFT Algorithm for OFDM Systems

Sangjin Ryoo

Department of Computer Information  
Hanyeong College University, 59720, Korea

Copyright © 2017 Sangjin Ryoo. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Abstract

In this paper, we analyze an implementation loss of a fast Fourier transform algorithm according to a bit truncation process. A 4K-point FFT algorithm is proposed and implemented in field programmable gate arrays. Because bit cannot be extended indefinitely, the truncation is required and performance varies depending on the truncation method in the implementation of the FFT algorithm. We measure the implementation loss according to a bit truncation process; the measured results show a better performance of truncation after FFT output.

**Keywords:** FFT, OFDM, bit truncation, quantization error, FPGA

## 1 Introduction

An orthogonal frequency-division multiplexing (OFDM) communication system is the structure generally used by many wired and wireless communications. A fast Fourier transform (FFT) algorithm is categorized by the way Cooley-Tukey recursive decomposition is applied. These decompositions finally reach butterfly operations, which greatly influences the FFT architecture. In an OFDM system, the FFT algorithm is very important, its performance being a crucial part of this system [1].

The study discussed in this paper used various fast algorithms for discrete Fourier transform (DFT) implementation, such as Cooley-Tukey, and DFT was implemented according to the Cooley-Tukey algorithm. Using this algorithm,

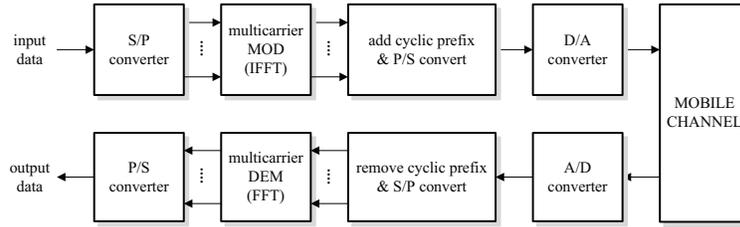


Figure 1: Block diagram of OFDM transmitter and receiver in mobile channel.

the architecture is modular and the 4K-point selection mode is implemented [2].

In this paper, we analyze the implementation loss of the FFT algorithm according to a bit truncation process. The first method involves the way to handle the truncation within stage. Truncation is processed by LSB truncation in the previous six stages and by MSB truncation in the next six stages. The second method involves the way to handle the truncation after FFT. We implement a FFT algorithm with field programmable gate array (FPGA) as a prototype. We also measure the implementation loss after bit truncation in a 4K-point FFT algorithm [3].

## 2 Structure of 4K-point FFT Algorithm including Truncation Block

The  $N$ -point discrete Fourier transform is defined by

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad k = 0, 1, \dots, N-1 \quad (1)$$

$$W_N = e^{-j(2\pi/N)} \quad (2)$$

where  $X(k)$  and  $x(n)$  are assumed to be complex numbers and  $W_N$  is the principal  $N$ th root of unity.

Figure 2 shows a comparison of the stage truncation and full truncation results using the same input data after IFFT/FFT processing. If there is no quantization error due to bit truncation, the input and output must be exactly the same, but their errors are not exactly the same as the result shown in Figure 2, because the stage truncation and the full truncation cause quantization errors. Figure 3 shows the implemented stage structure of a 4K-point FFT algorithm with truncation blocks. A 4K-point FFT algorithm consists of 12

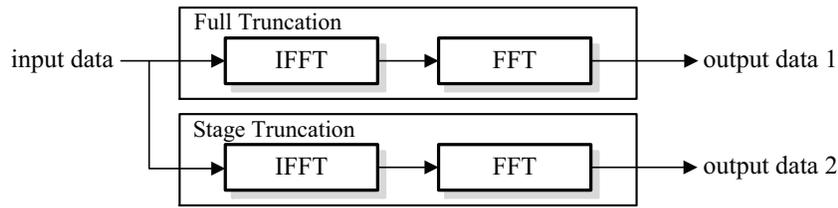


Figure 2: Input and Output Vector according to the Truncation Method.

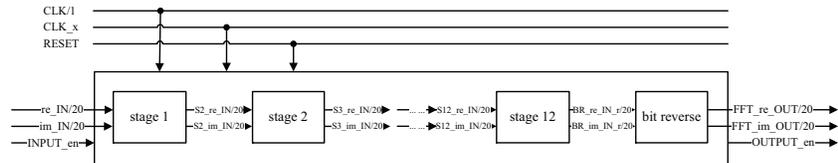


Figure 3: Structure of a 4K-point FFT Algorithm with Truncation Blocks.

stages, each of which operates in a radix-2 butterfly structure [4]. The processing capacity is reduced by half at all stages, and the first-stage processing capability and the second-stage processing capability are 4K and 2K, respectively, and so on [5]. Since all the stages contain truncation blocks, the output of all the stages is 20 bits. As shown in the Figure 3, since only one symbol is processed in the 12th stage, the multiplication block is eliminated [6],[7].

As previously mentioned, all stages are included in the truncation block in the bit truncation of a 4K-point FFT algorithm. The truncation block is used to control the number of bits processed at every step. The number of bits in the truncation block is controlled because the processing bits cannot be used indefinitely in a hardware implementation [8]. Based on the bit truncation operation, the performance of an FFT algorithm changes; thus, in the case of the MSB truncation method, the input and output are 21 and 20 bits, respectively, because one MSB bit operates for cancellation. In the case of the LSB truncation method, the input and output are 21 and 20 bits, respectively, because one LSB bit operates for removal [9],[10].

### 3 Structure of FFT Algorithm by Truncation after FFT

Figure 4 shows the implemented stage structure of a 4K-point FFT algorithm by truncation after the FFT. The truncation is processed after the FFT output, and all the bits are incremented by one at every step. As shown in Figure 4, one data symbol is processed in the twelfth stage, so the multiplication block is

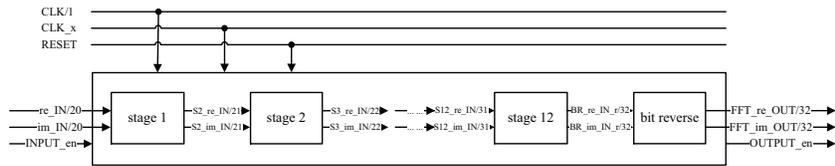


Figure 4: Structure of a 4K-point FFT Algorithm by the Truncation.

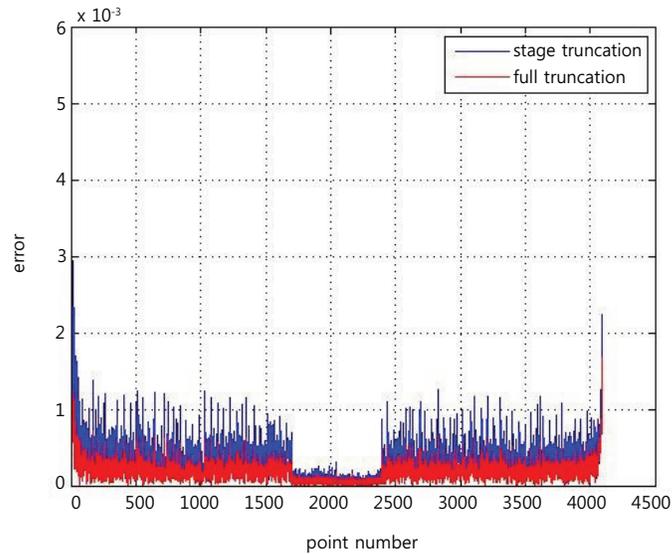


Figure 5: Implementation Loss of a 4K-point FFT Output According to Truncation Position.

removed. The truncation is also processed after the FFT operation; therefore, the truncation block is also removed from the stage block. The truncation process is used to control the number of bits processed in the bit truncation of a 4K-point FFT algorithm. Because the processing bits cannot be used indefinitely in a hardware implementation, the number of bits is controlled after the FFT operation. The bit truncation operation changes the performance of an FFT algorithm. This approach works by selecting only 20 bits of the full 32 bits of validity.

Figure 5 shows the implementation loss of a 4K-point FFT output due to the truncation position. Stage truncation means that the truncation is done at every step and that full truncation is performed after every step. Implementation loss is the difference between the input data and the output data after IFFT/FFT processing.

## 4 Conclusions

In this paper, we present an efficient design and implementation of a 4K-point FFT algorithm. The bit truncation process is adjusted to optimize hardware performance. The truncation process is used to control the number of processing bits in the bit truncation of the 4K-point FFT algorithm. The processing bit cant be used indefinitely for hardware implementation. Because of that, it is controlled the number of bits after FFT operation. The implemented 4K-point FFT algorithm was tested in a laboratory environment, and the test results showed the best performance of the truncation after the FFT output.

## References

- [1] J. W. Cooley and J. W. Tukey, An algorithm for machine calculation of complex Fourier series, *Mathematics of Computation*, **19** (1965), 297 - 301. <https://doi.org/10.2307/2003354>
- [2] F. Qureshi, S. A. Alam and O. Gustafsson, 4k-point FFT Algorithms Based on Optimized Twiddle Factor Multiplication for FPGAs, *2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia)*, (2010), 225 - 228. <https://doi.org/10.1109/primeasia.2010.5604921>
- [3] S. Cho and K. Kang, A Low-Complexity 128-Point Mixed-Radix FFT Processor for MB-OFDM UWB Systems, *ETRI Journal*, **32** (2010), 1 - 10. <https://doi.org/10.4218/etrij.10.0109.0232>
- [4] J. H. McClellan, R. W. Schafer and M. A. Yoder, *DSP FIRST: A Multimedia Approach*, Prentice Hall, 1988.
- [5] L. Hanzo, W. Webb and T. Keller, *Single- and Multi-carrier Quadrature Amplitude Modulation: Principles and Applications for Personal Communications, WATM and Broadcasting*, 2nd ed., IEEE Press-John Wiley, 2000.
- [6] S. He and M. Torkelson, A new approach to pipeline FFT processor, *Proceedings of International Conference on Parallel Processing*, (1996), 766 - 770. <https://doi.org/10.1109/ipps.1996.508145>

- [7] S. He and M. Torkelson, Designing pipeline FFT processor for OFDM (de)modulation, *URSI International Symposium on Signals, Systems, and Electronics*, (1998), 257 - 262. <https://doi.org/10.1109/issse.1998.738077>
- [8] J. E. Oh and M. S. Lim, New radix-2 to the 4th power pipeline FFT processor, *IEICE Trans. Electron.*, **E88-C** (2005), 1740 - 1746. <https://doi.org/10.1093/ietele/e88-c.8.1740>
- [9] F. Qureshi and O. Gustafsson, Low-complexity reconfigurable complex constant multiplication for FFTs, *IEEE Int. Symp. Circuits Syst.*, (2009), 1137-1140. <https://doi.org/10.1109/iscas.2009.5117961>
- [10] M. Hasan and T. Arslan, Scheme for reducing size of coefficient memory in FFT processor, *Electronics Letters*, **38** (2002), 163 - 164. <https://doi.org/10.1049/el:20020117>

**Received: February 3, 2017; Published: February 21, 2017**