Low Power FFT to Support OFDM Telecommunications in IoT Applications

Nikos Petrellis

Department of Electrical and Computer Engineering, University of Peloponnese, Patra, Greece; e-mail: npetrellis@uop.gr

Abstract. The minimum acceptable word length for the representation of real numbers in Fast Fourier Transform (FFT) with undersampled inputs is studied in this paper. FFT is a critical module in Orthogonal Frequency Division Multiplexing (OFDM) telecommunication infrastructure also used in Internet of Things (IoT) environments. The FFT input/output values, the twiddle factors and its intermediate results are complex numbers that can be represented either in fixed point or floating-point format. A tradeoff has to be made between rounding error and complexity. We focus on FFT with sparse inputs such as the values generated by many IoT sensors, surveillance cameras, etc. A configurable new FFT architecture has been developed to test various FFT sizes, word lengths and Quadrature Amplitude Modulations (QAM). The simulation results show that 8 bits FFT word length is sufficient in the employed undersampling procedure.

Keywords: FFT; sparse; OFDM; word length; rounding error.

1 Introduction

Discrete Fourier Transform (DFT) is used for the representation of periodic functions as a sum of sine and cosine functions. Fourier transform is used in several scientific domains of mathematics (e.g., for the analysis of differential equations) and signal processing (filtering, spectroscopy, etc.) [1]. Fast Fourier Transform (FFT) implementations avoid repeating the same operations [2]. One option is to implement real numbers in Floating Point format. This format supports wide dynamic range and avoids issues like scaling and overflow/underflow. In floating point format standards like IEEE-754 [3] a real number is described by one bit for the sign (*sgn*), a number of bits for the significand (*c*) and a signed exponent (*e*) for a base *b* such as 2 or 10. More specifically, the real number can be expressed as $(-1)^{sgn} \times c \times b^{e}$. Floating point format is implemented with complicated hardware. In Fixed Point format multiplications and divisions by 2 can be implemented with simpler circuits but the outcome of these operations may have scaling and overflow/underflow issues. In fixed point format, if for example, base b=2 is used, 8 bits are allocated for the whole number and 5 of the 8

Proceedings of the 9th International Conference on Information and Communication Technologies in Agriculture, Food & Environment (HAICTA 2020), Thessaloniki, Greece, September 24-27, 2020.

bits are used for the fraction then 10101101 corresponds to the real number: 1×22+0×21+1×20+0×2-1+1×2-2+1×2-3+0×2-4+1×25=4+1+1/4+1/8+1/32= 5.40625.

The representation of the real numbers is very important to computationally intensive operations like FFT. Due to the FFT's high complexity, the real numbers have to be implemented preserving a minimum word length, otherwise severe rounding errors occur. The limited precision of fixed-point arithmetic for different FFT algorithms is studied in [4] where radix-2 Decimation-In-Time (DIT) FFT is examined due to its higher accuracy in term of signal-to-quantization-noise ratio. In [5] the round-off error of fixed point FFT is investigated while the results of the classic paper [6] are reproduced and an error in the results presented in [6] is demonstrated. The consequences of the violation of the assumption for almost pure sine waves is also investigated in [5]. Radix-4 FFT is used in [7] where input quantization and coefficient accuracy is ignored. The error in Radix-2 butterflies is analyzed in [8].

The analysis for the FFT word length presented here concerns an OFDM transceiver that has been recently presented in [9]. In this OFDM transceiver undersampling is applied when sparse information is exchanged i.e., some FFT input samples are not obtained by the Analog Digital Converter (ADC) on the receiver side and are replaced by others that have been previously retrieved. A configurable FFT has been developed in synthesizable Very high-speed IC Hardware Description Language (VHDL) in the context of this paper in order to test the effect of limited word length in the representation of the FFT parameters. These parameters include inputs, outputs, twiddles and the intermediate results. The following combinations were studied here: FFT size *N* of 256 or 1024 points, q=16 or q=4 QAM modulation, two subsampling frequencies (R2=N/4 and R8=N/16) and sparseness levels up to 10%. Fixed point word lengths of 8 or 6 bits are tested. The Normalized Mean Square Error (NMSE) and the Symbol Error Rate (SER) are measured. The NMSE and SER are severely degraded if 6 bits are used. However, if 8 bits are used as FFT word length, the quantization error is negligible compared with the one caused by the undersampling procedure.

The OFDM and FFT architectures as well as the proposed undersampling method are described in Section 2. The simulation results are discussed in Section 3.

2 OFDM and FFT Architectures

In [9], wired and wireless OFDM transceivers are described and undersampling is supported during sparse information exchange: some input samples of the receiver FFT can be omitted and replaced by others. In this work, we use the wired OFDM transceiver model in order to study the rounding effect in the FFT. The input of the OFDM transmitter is encoded generating a parity bit stream. If q-QAM modulation is used, $\log_2(q)$ bits from the interleaved parity/data bit streams are mapped to the corresponding constellation symbol X_k ($0 \le k < N$). At the input of the N-point IFFT, q-QAM symbols are arranged in a proper order and the symbols x_n ($0 \le n < N$) are generated at the IFFT output. The reverse procedure is followed on the receiver where the symbols $y_n = x_n + z_n$ form the N-pointt FFT input. The symbol z_n is Additive White Gaussian Noise (AWGN) noise. The output of the FFT are N, Y_k symbols. An appropriate error decoder such as Viterbi corrects a number of errors using the parity bits, avoiding packet retransmission

The parameters y_n, Y_k are the N, DFT input, output symbols and X_k, x_n are the N, Inverse DFT (IDFT) input, output symbols respectively. The twiddle factors w are defined as $w_N^r = e^{2\pi r/N}$. The x_n symbols of the OFDM transmitter IDFT output are sent over the communication channel and they are received as y_n at the input of the receiver DFT module. FFT reduces the $O(N^2)$ operations required by DFT to $O(N \cdot \log N)$. FFT building blocks are the Radix-r butterflies. Radix-2 butterflies operate on a pair of inputs (y_1 and y_2) that are defined as follows [10] for Decimation in Time (DIT) FFT:

$$Y(2m) = \sum_{n=0}^{\frac{N}{2}-1} (y_{2m\_a}(n) + y_{2m\_b}(n)) w_{N/2}^{mn}. \quad \textbf{(1)}$$

$$Y(2m+1) = \sum_{n=0}^{\frac{N}{2}-1} (y_{(2m+1)\_a}(n) - y_{(2m+1)\_b}(n)) w_{N/2}^{kn}. \quad \textbf{(2)}$$

The FFT input y_k is split in two parts: y_{k_a} and y_{k_b}. In Radix-4 butterflies, 4 pairs of inputs are used and radices that are not powers of 2 (such as Radix-3, Radix-5) can also be employed. Decimation in Frequency (DIF) differs from DIT in the fact that the order of the butterfly inputs and outputs is bit reversed. The FFT software implementations are slow and thus, most of the modern telecommunication systems are based on hardware implementations.

There are $\log_2(N)$ stages in the N-point FFT implemented in this paper. The real and imaginary parts of the stage p inputs are stored in a pair of buffers. A pair of operands are accessed through ports r_{p1} and r_{p2} while a pair of values can be stored to the buffer through write ports w_{p1}, w_{p2}. Each one of these read or write ports has size d. The twiddle factors w, are retrieved from a ROM with $N/2^{p+1}$ size at stage p. Each Butterfly block performs the following calculations:

$$Re\{w_{(p-1)1}\} = Re\{r_{p1}\} + Re\{r_{p2}\} \cdot Re\{tw\} - Im\{r_{p2}\} \cdot Im\{tw\}. \quad \textbf{(3)}$$

$$Im\{w_{(p-1)1}\} = Im\{r_{p1}\} + Re\{r_{p2}\} \cdot Im\{tw\} + Im\{r_{p2}\} \cdot Re\{tw\}. \quad \textbf{(4)}$$

$$Re\{w_{(p-1)2}\} = Re\{r_{p1}\} - Re\{r_{p2}\} \cdot Re\{tw\} + Im\{r_{p2}\} \cdot Im\{tw\}. \quad \textbf{(5)}$$

$$Im\{w_{(p-1)2}\} = Im\{r_{p1}\} - Re\{r_{p2}\} \cdot Im\{tw\} - Im\{r_{p2}\} \cdot Im\{tw\}. \quad \textbf{(6)}$$

The sparseness level S is the fraction of the non-zero bits in the input data stream and is expected to be small (e.g., 1%-2%) if the input data are sparse. Many q-QAM symbols are expected to be equal (X_c) due to data sparsity and they form an IFFT input set with appropriate structure. A number of samples in this FFT input set are expected to be equal (due to the input sparseness) and they can replace each other. The ADC in does not need to sample all the values since some of them can be replaced by others already received. The ADC sampling rate can thus be reduced on the receiver for lower power. The OFDM undersampling method proposed for wired channels is based on the following DFT property:

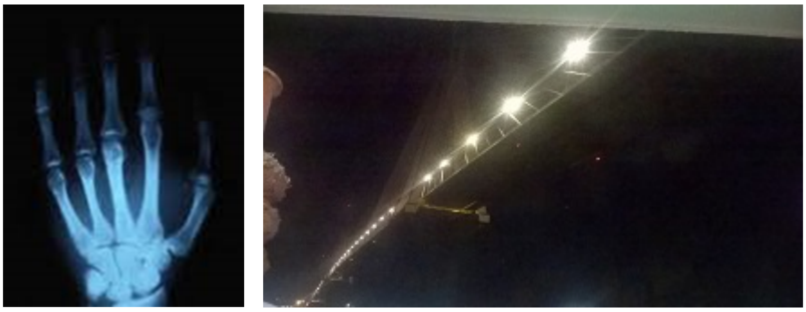
$$x_n = \frac{1}{N} \left(\sum_{k=0,2,..}^{N-2} X_k w_N^{kn} + \sum_{k=1,3..}^{\frac{N}{2}-1} (X_k - X_{k+\frac{N}{2}}) w_N^{kn} \right). \quad \textbf{(7)}$$

$$x_{n+\frac{N}{2}} = \frac{1}{N}\left(\sum_{k=0,2,}^{N-2} X_k w_N^{kn} - \sum_{k=1,3,}^{\frac{N}{2}-1} (X_k - X_{k+\frac{N}{2}}) w_N^{kn}\right). \quad \textbf{(8)}$$

The outputs x_n and $x_{n+\frac{N}{2}}$ of the IDFT are equal, only if $X_k = X_{k+\frac{N}{2}}$ and k is odd. In this case $x_{n+\frac{N}{2}}$ (and thus: $y_{n+\frac{N}{2}}$) can be replaced by x_n (y_n). On the receiver, up to half of the y_n symbols placed in odd positions with n>N/2, can be substituted by others. The ADC on the receiver can operate at half speed, 50% of the time in this case. The maximum number of substituted samples is R2=N/4 but this value would make sense only if all inputs are 0, otherwise a high error floor occurs. If the value of R is smaller, the error may be acceptable.

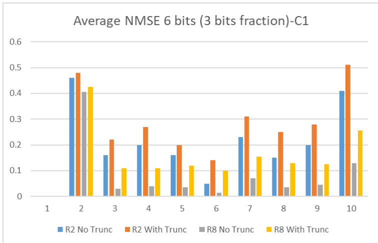
4 Simulation

The effect of the rounding error caused by the limited FFT word length is studied through simulations. The two metrics used, are the Normalized Mean Square Error (NMSE) and the Symbol Error Rate (SER). If it is assumed that an FFT output Y corresponds to IFFT input X, its NMSE error can be expressed as $\varepsilon = \|Y - X\|_2^2 / \|X\|_2^2$. There is an area on the QAM constellation plane where Y is recognized as X due to shorter distance compared with the neighboring constellations. Although the NMSE is not 0 in this case this error does not affect SER. If e.g., Y is recognized as a different 16QAM symbol than X, only one of the 4 data bits that correspond to this constellation would be inverted. Thus, the effect of this error on the Bit Error Rate (BER) would be 4 times lower than on the SER.

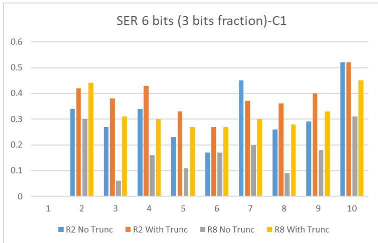


(a) (b)

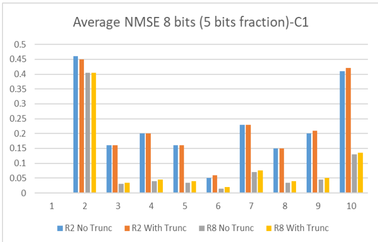
Fig. 1. Sparse images used to define the IFFT input structures tested in this paper.



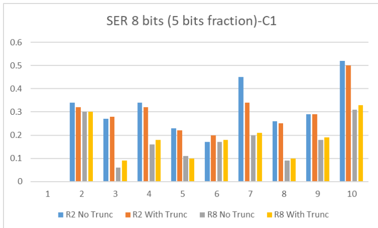
(a)



(b)



(c)



(d)

Fig. 2. Combination C1. NMSE (a), SER (b) with 6 and NMSE (c), SER (d) with 8-bits word.

If two numbers of n_b bits are multiplied, the rounding error is between $\pm\frac{2^{-(n_b-1)}}{2}$. If $d_s = 2^{-(n_b-1)}$ and if the error probability is uniform ($1/d_s$), then the error variance is: $d_s^2/12$. The rounding noise for N-point FFT is up to [11]: $P_{RE} = \frac{d_s^2}{6}(log_2 N - 2)$. The undersampling error for 16QAM as explained in [9] is $P_{UE} = (R \cdot S \cdot \sqrt{2}(\sqrt{q} - 1)log_2 q)^2$. The simulation results presented in this section highlight the cases where selecting appropriate number of bits n_b makes $P_{RE} << P_{UE}$. Three representative combinations are examined: C1) N=256, 16QAM, C2) N=256, 4QAM (Quadrature Phase Shift Keying-QPSK) and C3) N=1024, 16QAM. Data are taken from sparse images like the ones shown in Fig. 1. The IFFT input packets formed have sparseness *S* between 5% and 30%. This is too high for the proposed undersampling procedure as explained in [9] but these IFFT input packets will merge with several others consisting of only X_c symbols (S=0). Thus, a projection is also used to display how NMSE and SER changes according to *S*.

Fig. 2 shows the simulation results for combination C1. When 6 bits (with 3 bits fraction) are used, truncation ("Trunc") caused by limited word length plus undersampling error makes the quantization error much higher than the undersampling error alone ("NoTrunc" cases). However, the error is almost identical and equal to the undersampling error when 8 bits (with 5 bits fraction) are used i.e., the quantization error is small in this case.

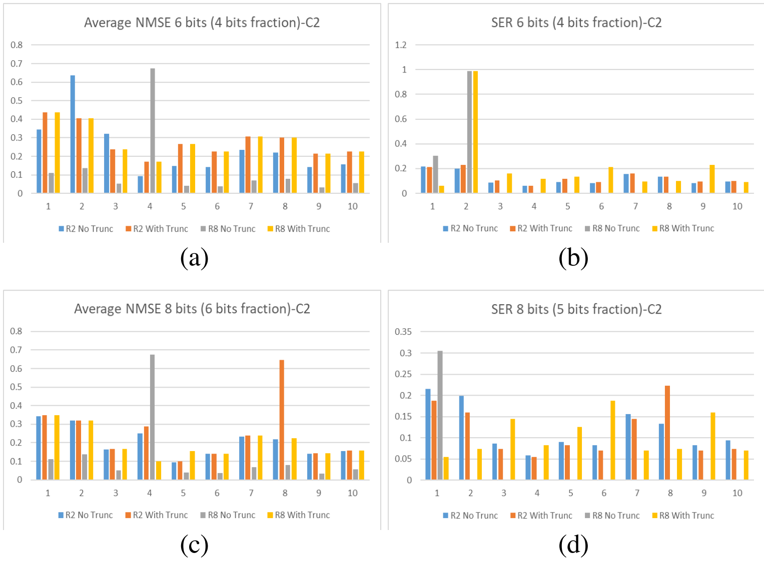


Fig. 3. Combination C2. NMSE (a), SER (b) with 6 and NMSE (c), SER (d) with 8-bits word.

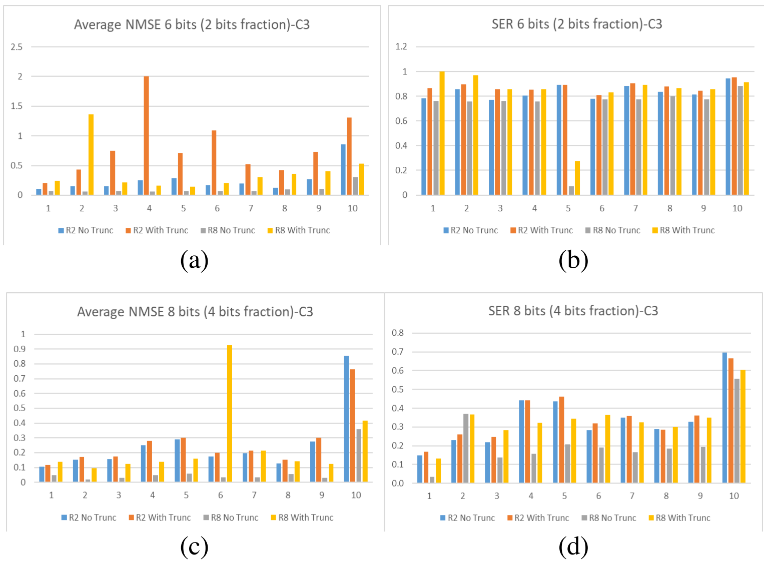
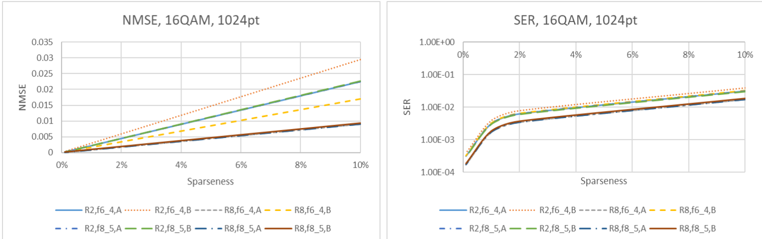


Fig. 4. Combination C3. NMSE (a), SER (b) with 6 and NMSE (c), SER (d) with 8-bits word.

Similar conclusions can be drawn from Fig. 3 and 4 for combinations C2 and C3, respectively. It should be noticed that the undersampling error is quite small with QPSK (C2) especially with R8, making the quantization error relatively high even when 8-bit word length is used. In the C3 combination, different NMSE values lead to similar SER behavior. NMSE can be drastically reduced when 8-bit word length is selected. Fig. 5 shows how the NMSE, SER can be projected in the range S=[0..10%] for one of the cases (C3). Similar plots can be retrieved for C1 and C2.



(a) (b)

Fig. 5. NMSE (a) and SER (b) projection for C3 in the range S=[0..10%].

5 Conclusions

The effect of the selected FFT word length in an OFDM transceiver supporting undersampling when sparse information is exchanged as in the case of IoT sensors, surveillance cameras, etc. was studied. Simulation on three combinations of FFT size and QAM modulations, showed that the use of 8-bits in the FFT word length was adequate in order to keep the power consumption and complexity low.

References

1. Bracewell, R. (1999) The Fourier Transform & Its Applications. 3rd ed. McGraw-Hill.
2. Cooley, J.W. and Tukey, J.W. (1965) An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematical Computer, 19, p. 297-301.
3. IEEE Standard for Floating Point Arithmetic, ANSI/IEEE Standard 754/2008, Aug. 2008.
4. Chang, W-H and Nguyen, T. (2008) On the Fixed-Point Accuracy Analysis of FFT Algorithms. IEEE Trans. On Signal Processing, 56(10), p. 4673-4682.
5. Pálfi, V. and Kollár, I. (2009) Roundoff Errors in Fixed-Point FFT. Proc. of the IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary.
6. Welch, P.D. (1969) A fixed-point fast Fourier transform error analysis. IEEE Transactions on Audio and Electroacoustics, 17(2), p. 151-157.
7. Prakash, S. and Rao, V. (1981) Fixed-Point Error Analysis of Radix-4 FFT. Signal Processing, North Holland Publishing Company, 3, p. 123-133.
8. Liu, B. and Thong, T (1976) Fixed-Point Fast Fourier Transform Error Analysis. IEE Trans. Acoustics, Speech, Signal Processing, ASSP-24, p. 563-573.

9. Petrellis, N. (2016) Optimal Reconstruction of Sub-sampled Time-Domain Sparse Signals in Wired/Wireless OFDM Transceivers. Springer EURASIP Journal on Wireless Communications and Networking, 122.
10. Löfgren, J. and Nilsson, P. (2011) On hardware implementation of radix 3 and radix 5 FFT kernels for LTE systems. Proc of the IEEE NORCHIP conference, Lund, Sweden.
11. Widrow, B. and Kollár, I. (2008) Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications. Cambridge University Press, Cambridge, UK.