Selecting FFT Word Length in OFDM systems with Undersampled Sparse Input

Nikos Petrellis
Electrical and Computer Engineering Dept.
University of Peloponnese
Patras, Greece
npetrellis@uop.gr

***Abstract*— A critical module in telecommunication applications that are based on Orthogonal Frequency Division Multiplexing (OFDM) is the Fast Fourier Transform (FFT). Several FFT implementations have been proposed in the last 50 years aiming at the reduction of the required resources. The rounding error caused by the limited word length for the various FFT parameters has been extensively studied for many FFT implementation alternatives. In this paper, the minimum acceptable FFT word length is investigated for a specific OFDM architecture where undersampling is applied at the receiver when sparse data are exchanged. In the context of this paper, a configurable new FFT architecture has been developed in hardware description language in order to test various FFT sizes, word lengths and Quadrature Amplitude Modulation (QAM) levels. The simulation results show that the error floor caused by the undersampling in the specific OFDM system makes unnecessary the use of more than 8 bits as word length when 256-pt or 1024-pt FFT and 16QAM or 4QAM is used.**

Keywords—FFT, sparse, OFDM, word length, rounding error

I. INTRODUCTION

One way to implement real numbers is the use of floating point format that allows wide dynamic range, freeing special purpose processor designers from the scaling and overflow/underflow concerns [1]. The IEEE-754 standard 32-bit floating-point format [2] can also be used for the communication between a FFT and a general purpose processor. In floating point formats like IEEE-754 a real number is described by one bit for the sign (*sgn*), *c*-bits for the significand and a signed exponent (*e*) for a base *b* such as 2 or 10. The resources needed to handle numbers expressed in floating point format is relatively high. Another way to represent real numbers is the use of fixed point format. Several operations such as multiplications and divisions by 2 can be implemented with simpler circuits in fixed point format but the results of these operations have to be monitored for scaling and overflow/underflow issues. We focus on fixed point format in this paper.

The FFT parameters and its input/output values cannot be implemented with an arbitrary small word length, because severe rounding errors may occur. The limited precision of fixed-point arithmetic for different FFT algorithms is studied in [3] where radix-2 Decimation-In-Time (DIT) FFT is used due to its higher accuracy in term of signal-to-quantization-noise ratio. In [4] the round-off error of fixed point FFT is investigated while the results of the classic paper [5] are attempted to be reproduced in [4]. The effect of rounding is also examined in [6] and [7]. Radix-4 FFT is examined in [6]. Input quantization and coefficient accuracy is not taken into consideration. Analytical expressions are derived for DIT and Decimation-In-Frequency (DIF) FFTs. Radix-2 error effects are analyzed in [7].

An OFDM transceiver where undersampling is applied when sparse information is exchanged, has been recently described in [8] and [9]. The Bit Error Rate (BER) as a function of the channel SNR for various OFDM configurations (FFT points, QAM modulation used, etc) has been presented. Although the round-off error has been modeled, it was not taken into consideration in the simulation results presented in [8] and [9]. In this paper, a configurable FFT has been developed in synthesizable Very high speed IC Hardware Description Language (VHDL) in order to test the effect of limited precision in the representation of the FFT parameters (inputs, outputs, twiddles, intermediate results). More specifically, 256 and 1024-point FFT with 16QAM or Quadrature Phase Shift Keying (QPSK) modulation is studied. The Normalized Mean Square Error (NMSE) and the Symbol Error Rate (SER) are measured for 8 and 6-bit fixed point word lengths. The simulation results show that the use of a 6-bit word length degrades significantly NMSE and SER while their degradation is small if 8 bits are used since the error caused by the employed undersampling procedure is dominant in this case. These effects are studied for several levels of input sparsity. The methodology and tools presented in this paper can be used to further explore the effects of the round-off errors in various FFT configurations (different FFT size, QAM modulations, etc).

This paper is structured as follows: In Section II the fundamentals of FFT architecture are summarized. The developed FFT architecture and the OFDM undersampling method presented in [8] and [9] is briefly described in Section III. Section IV presents the simulation results for the two fixed point word lengths examined in this paper.

II. FFT ARCHITECTURE

The Discrete Fourier Transform (DFT) and the Inverse DFT (IDFT) are defined by eq. (1) and (2) respectively:

$$Y_k = \sum_{n=0}^{N-1} y_n w_N^{-kn}, 0 \le k < N-1 \quad (1)$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k w_N^{kn}, 0 \le n < N-1 \quad (2)$$

where y_n, Y_k are the N, DFT input, output symbols and X_k, x_n are the N, IDFT input, output symbols respectively. The twiddle factors w are defined as $w_N^r = e^{2\pi r/N}$. Different symbols are used (y_n, x_n) in eq. (1) and (2), because the x_n symbols of the IDFT output of the OFDM transmitter are sent over the communication channel, the noise modifies them 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 can be implemented using interconnected building blocks called Radix-r butterflies. The simplest case are the Radix-2 butterflies. They operate on a pair of inputs y

and are defined by the following relations [10] for Decimation in Time (DIT) FFT:

$$Y(2k) = \sum_{n=0}^{\frac{N}{2}-1}(y_1(n) + y_2(n))w_{N/2}^{kn} \quad (3)$$

$$Y(2k+1) = \sum_{n=0}^{\frac{N}{2}-1}(y_1(n) - y_2(n))w_{N/2}^{kn} \quad (4)$$

The y_1 and y_2 vectors are the two halves of the FFT input vector y. In Decimation in Frequency (DIF), the order of the butterfly inputs and outputs is bit reversed.

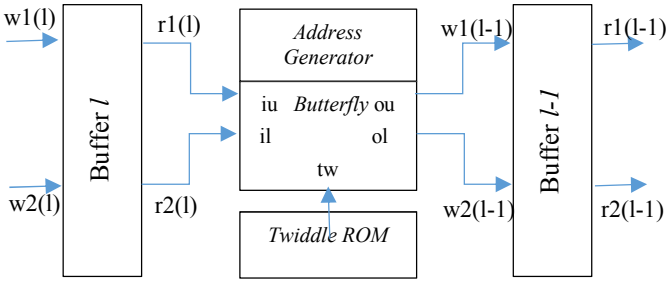

Fig. 1. One stage of the developed FFT.

Hardware FFT implementations can achieve low latency and can either consist of a large number of operators working in parallel or reusable components that lead to lower cost/power circuits but show a slightly higher latency.

The FFT implemented in this paper consists of log(N) stages like the one shown in Fig. 1. The inputs of stage l are stored in the double buffer l. One buffer stores the real and the other the imaginary parts of the FFT inputs/outputs. Henceforth, we refer to Buffer l, as if it was a single buffer and each one of the ports below refers to a complex number. Buffer l is accessed for write through ports w(l) (w1(l) and w2(l)) and for read through ports r(l) (r1(l) and r2(l)). These ports consist of an address bus (ra(l) or wa(l)) of size $\log_2 N$ bits and a pair of data buses (Re{rd(l)} and Im{rd(l)})with size d. Although the size of the butterfly outputs can gradually increase by one bit in every stage for optimal resource utilization we select a constant d size for all stages for simplicity. The data bus carries real numbers in fixed point format with a size of d bits. The inputs of each Radix-2 butterfly are the rd1(l) and rd2(l) while its outputs are wd1(l) and wd2(l). The real and imaginary parts of the twiddle factors w are retrieved from the twiddle ROM. The size of the twiddle ROM of stage l is $N/2^{l+1}$. The operations performed by the Butterfly block are derived by eq. (3) and eq. (4):

$$Re\{ou\} = Re\{iu\} + Re\{il\} \cdot Re\{tw\} - Im\{il\} \cdot Im\{tw\} \quad (5)$$

$$Im\{ou\} = Im\{iu\} + Re\{il\} \cdot Im\{tw\} + Im\{il\} \cdot Re\{tw\} \quad (6)$$

$$Re\{ol\} = Re\{iu\} - Re\{il\} \cdot Re\{tw\} + Im\{il\} \cdot Im\{tw\} \quad (7)$$

$$Im\{ol\} = Im\{iu\} - Re\{il\} \cdot Im\{tw\} - Im\{il\} \cdot Im\{tw\} \quad (8)$$

The address buses ra(l) and wa(l) are driven by the Address Generator module. It is based on an up counter with $\log_2(N)$ resolution and the modulo of this counter value with different dividers, generates the correct Buffer addresses for accessing the operands and the results of the Butterflies.

III. OFDM Undersampling Method

In [8] and [9], wired and wireless OFDM transceivers are described where 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. Special pilot symbols are placed on reserved subcarriers for channel estimation and equalization. The reverse procedure is followed at 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 employed undersampling method is exploiting the sparseness of the input data. Let S denote the fraction of the non-zero bits in the input data stream. If data are sparse, S is expected to be small e.g., 0.5%-10%. The sparsity of the images used as case studies in [8] and [9] is within this range. Many q-QAM symbols are likely to have identical value due to the data sparseness and are arranged appropriately at the input of the IFFT. If $z_n=0$ and no undersampling is applied, then $x_n=y_n$ and $X_k=Y_k$. However, if z_n is small but not zero, it is still expected that $X_k \approx Y_k$. Higher z_n causes the same effect as using input data with worse sparsity. A number of samples at the FFT input are replaced by others that have already been received. This substitution of samples is allowed due to the data sparseness and certain DFT properties. The ADC sampling rate can be reduced on the receiver for lower power. The Interleaver in the proposed scheme, generates q-QAM symbols, either from parity, or data bits only using a small buffer at the output of the encoder in order to store $2\log_2(q)$ data and parity bits. The OFDM undersampling method proposed for wired channels is based on the following DFT property:

$$x_n = \frac{1}{N}\left(\sum_{k=0,2,..}^{N-2} X_k w_N^{kn} + \sum_{k=1,3..}^{\frac{N}{2}-1}(X_k - X_{k+\frac{N}{2}})w_N^{kn}\right) \quad (9)$$

$$x_{n+\frac{N}{2}} = \frac{1}{N}\left(\sum_{k=0,2,}^{N-2} X_k w_N^{kn} - \sum_{k=1,3,}^{\frac{N}{2}-1}(X_k - X_{k+\frac{N}{2}})w_N^{kn}\right) \quad (10)$$

According to eq. (9) and (10), the IDFT outputs x_n and $x_{n+\frac{N}{2}}$ are equal, only if X_k and $X_{k+\frac{N}{2}}$, are equal (when k is odd). If the X_k and $X_{k+\frac{N}{2}}$ symbols (data, pad or pilots) are identical ($=X_c$) then $x_{n+\frac{N}{2}}$ (and thus: $y_{n+\frac{N}{2}}$) can be replaced by x_n (or y_n respectively). This symbol equivalence stands for most of the data symbols if the input is sparse and if the pad and pilot symbol values are selected equal to X_c. Up to half of the y_n symbols (with odd $n>N/2$), on the receiver side can be replaced by others and consequently, up to 50% of the time, the receiver's ADC sampling rate can be reduced to

the half (undersampling mode). The number of substituted y_n samples is denoted here using the letter R (R≤N/4). The maximum value for R (R2=N/4) can be used only if the input is constant (e.g., 0). An R value lower than N/4 has to be selected in order to achieve an acceptable error. The appropriate value for R depends on the sparseness level of the input data.

IV. Effect of FFT Round-off Error

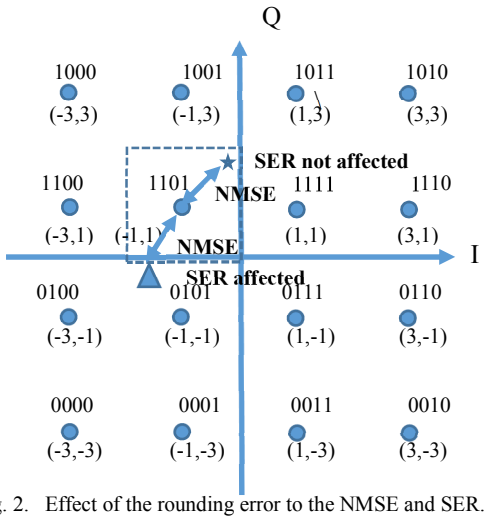

Fig. 2. Effect of the rounding error to the NMSE and SER.

In this section, the truncation effect of the round-off error caused by the limited precision of the real numbers' implementation in fixed point format is studied. The Normalized Mean Square Error (NMSE) and the Symbol Error Rate (SER) are the two metrics used to quantify the rounding error. Fig. 2 shows the constellations for 16QAM modulation. Two FFT outputs Y_{1a} and Y_{1b} (denoted by a star and a triangle, respectively) on the OFDM receiver are shown in Fig. 2. If we assume that both of these FFT outputs correspond to the same IFFT input X_1=(-1,1) on the OFDM transmitter, then the NMSE of these two FFT outputs ε_{1a} and ε_{1b}, measures in a sense, the normalized distance between the correct constellation value X1 and these two FFT outputs: $\varepsilon_{1a} = \|Y_{1a} - X_1\|_2^2 / \|X_1\|_2^2$ and $\varepsilon_{1b} = \|Y_{1b} - X_1\|_2^2 / \|X_1\|_2^2$. The dashed rectangle shows the area where an FFT output is recognized as X_1 due to shorter distance from the neighboring constellations. Although $\varepsilon_{1a} \neq 0$, Y_{1a} (star) does not degrade the SER since it resides within the dashed rectangle and will be recognized as X_1. On the contrary, Y_{1b} (triangle) degrades the SER since it will be recognized as X_2=(-1,-1) (it is closer to this constellation) rather X_1. Had Bit Error Rate (BER) been measured, the error would appear to only one of the constellation four bits, thus the effect on the BER of this error is 1/4 of the effect on the SER. Thus, SER represents the worst case effect of the error to the OFDM system reception quality.

The round-off error of the multiplication of two numbers with size n_b bits, is between $-\frac{2^{-(n_b-1)}}{2}$ and $\frac{2^{-(n_b-1)}}{2}$. The distance is $d_s = 2^{-(n_b-1)}$ between these limits and if the error probability is assumed to be uniform ($1/d_s$) between $-d_s/2$ and $d_s/2$, then the variance of the error is estimated as: $d_s^2/12$. The round off noise for N-point FFT implementations can be up to [8][11]: $\frac{d_s^2}{6}(log_2 N - 2)$. The power of the undersampling error is equal to [8]: $(R \cdot S \cdot \sqrt{2}(\sqrt{q} - 1) log_2 q)^2$ for 16QAM modulation. This means that higher error is expected for big R, S and q values. In the following simulations we used three combinations of FFT size and QAM modulation: a) 16QAM (q=16), N=256 points FFT, b) QPSK (q=4), N=256 and c) 16QAM, N=1024. Two cases are examined for R: R2=N/4 and R8=N/16 (the number following symbol R corresponds to the distance of the samples that will be omitted). Table I shows the average NMSE and SER estimated using 10 FFT inputs with sparseness ranging from 4.5% to 31% for each FFT input. In the case of 16QAM and N=256 the fixed point formats *f6_3* (and *f8_5*) have been tested, meaning that 3 (or 5) of the 6 (or 8) word length bits have been used for the fraction. A slightly different number of fraction bits is used in the rest of the QAM and N parameter combinations that have been tested.

The error measured from Octave corresponds to the undersampling effect only and it is compared with the error measured from the VHDL/Modelsim where the developed FFT has been described. The error in the latter case is affected by both the undersampling and bit truncation. The first case (Octave) is denoted by "NoTr" in the following figures and tables. The second case (Modelsim) is denoted by "Tr". As can be seen in all cases of Table I, both the NMSE and SER errors get quite close (whether the bit truncation error is taken into consideration or not), if 8-bits are used as word length. If 6-bits are used, the error in the Tr case is much higher (double in some combinations) than the error achieved in the NoTr case.

TABLE I. Average NMSE and SER

16QAM N=256	***R2, f6_3 (NoTr, Tr)***	***R8, f6_3 (NoTr, Tr)***	***R2, f8_5 (NoTr, Tr)***	***R8, f8_5 (NoTr, Tr)***
NMSE	0.244, 0.295	0.089, 0.17	0.244, 0.226	0.089, 0.093
SER	0.318, 0.386	0.175,0.327	0.318, 0.302	0.175, 0.186
QPSK N=256	***R2, f6_4 (NoTr, Tr)***	***R8, f6_4 (NoTr, Tr)***	***R2, f8_6 (NoTr, Tr)***	***R8, f8_6 (NoTr, Tr)***
NMSE	0.243, 0.261	0.131, 0.261	0.243, 0.244	0.131, 0.183
SER	0.109, 0.120	0.110, 0.235	0.109, 0.106	0.110, 0.104
16QAM N=1024	***R2, f6_2 (NoTr, Tr)***	***R8, f6_2 (NoTr, Tr)***	***R2, f8_4 (NoTr, Tr)***	***R8, f8_4 (NoTr, Tr)***
NMSE	0.275, 0.855	0.102, 0.411	0.275, 0.284	0.102, 0.259
SER	0.363, 0.876	0.240, 0.814	0.363, 0.377	0.240, 0.361

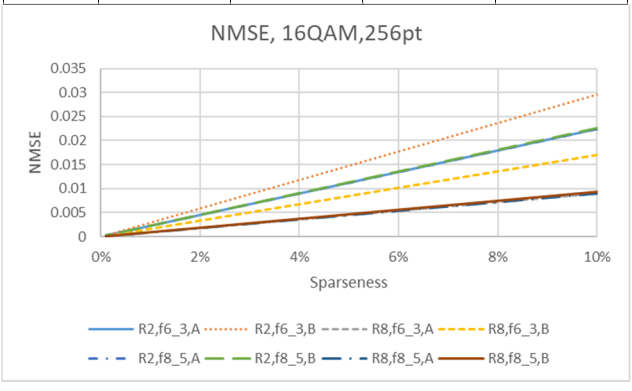

Fig. 3. NMSE for 16QAM and 256-pt FFT at sparseness levels up to 10% combining f6_3 or f8_5, R2 or R8 and NoTr (A) or Tr (B).

Using the maximum value for R (R2=1/4), the lowest sampling rate can be applied leading to the lowest power consumption. Nevertheless, the undersampling procedure poses a higher error in this case. Using R8, a higher power is required since only N/16 samples are omitted by the sampling

procedure of the ADC. However with R8, the undersampling error is lower and acceptable for many applications.

In order to demonstrate the average NMSE of a larger input stream instead of individual FFT inputs, a projection is made by taking into account the fraction of trivial data in the input stream. Fig. 3 shows the average NMSE for sparseness levels up to S=10%. As expected the smaller NMSE is achieved when *f8_5* and R8 is used. The plots are similar for the other QAM and N parameter combinations. The SER displayed in Table I is too high in most of the cases, but this is owed to the fact that it is derived by FFT inputs that are not sparse. Fig. 4 shows a projection of SER similar to the one displayed for NMSE in Fig. 3 for N=1024 and 16QAM. As can be seen from Fig. 4 a SER close to 10^{-4} can be achieved if the sparseness level *S* is less than 1%. As already explained at the beginning of this section BER is expected to be even lower.

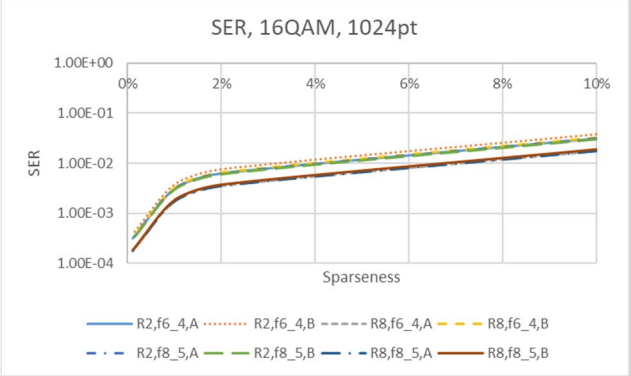

Fig. 4. SER for 16QAM and 1024-pt FFT at sparseness levels up to 10% combining f6_3 or f8_5, R2 or R8 and NoTr (A) or Tr (B).

The developers are often adopting 16 or 32-bits for the implementation of the real numbers avoiding rounding errors in computationally intensive blocks like FFTs. However, it was shown by the simulation results presented here, that a smaller number of bits can be used in applications where a higher error floor is caused by other sources like undersampling. A screenshot of the Modelsim simulation environment is shown in Fig. 5. The *d_re*/*d_im* pair of signals at the top is the FFT input and the next pair of signals (*q_re*/*q_im*) is the FFT output.

V. Conclusions

The effect of the round-off errors caused by the implementation of real numbers with fixed point format in an OFDM transceiver that supports undersampling, has been studied in this paper. A robust configurable FFT architecture has been developed on the OFDM receiver side. The NMSE and SER error have been measured when a number of FFT inputs has been substituted by others during sparse information exchange. The results showed that for all the FFT parameter combinations tested, the real numbers do not need to be implemented with more than 8 bits due to the error floor caused by the employed undersampling process.

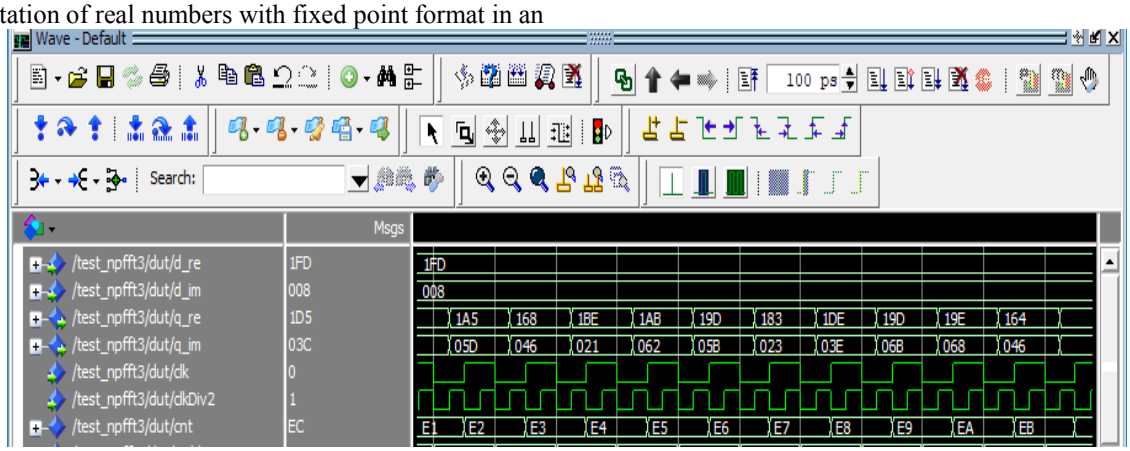

Fig. 5. The Modelsim 10.3.c simulation environment for the 256 pt FFT developed in VHDL.

Acknowledgment

This work is protected by the provisional patents 1008564 (published Sep. 9, 2015) and 1008130 (published March 6, 2014), Greek Patent Office (OBI).

References

[1] E. Swartzlander Jr., H. Saleh, "FFT Implementation with Fused Floating-Point Operations," IEEE Trans. On Computers, vol. 61, no. 2, pp. 284-288, Feb. 2012.

[2] IEEE Standard for Floating Point Arithmetic, ANSI/IEEE Standard 754/2008, Aug. 2008.

[3] W-H Chang and T. Nguyen, "On the Fixed-Point Accuracy Analysis of FFT Algorithms," IEEE Trans. On Signal Processing, vol. 56, no. 10, pp. 4673-4682, Oct. 2008.

[4] V. Pálfi and I. Kollár, "Roundoff Errors in Fixed-Point FFT," Proc. of the IEEE International Symposium on Intelligent Signal Processing, 26-28 Aug. 2009, Budapest, Hungary.

[5] P.D. Welch, "A fixed-point fast Fourier transform error analysis," IEEE Transactions on Audio and Electroacoustics, vol. 17, no. 2, pp. 151-157, 1969.

[6] S. Prakash and V. Rao, "Fixed-Point Error Analysis of Radix-4 FFT," Signal Processing, North Holland Publishing Company, vol. 3, pp. 123-133, 1981.

[7] B. Liu and T. Thong, "Fixed-Point Fast Fourier Transform Error Analysis," IEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-24, pp. 563-573, December 1976.

[8] N. Petrellis, "Optimal Reconstruction of Sub-sampled Time-Domain Sparse Signals in Wired/Wireless OFDM Transceivers", Springer EURASIP Journal on Wireless Communications and Networking 2016:122.

[9] N. Petrellis, "Low Power OFDM Receiver Exploiting Data Sparseness and DFT Symmetry," Hindawi, International Journal of Distributed Sensor Networks, vol. 2016, Article ID 1464639.

[10] J. Löfgren and P. Nilsson, "On hardware implementation of radix 3 and radix 5 FFT kernels for LTE systems", Proc of the IEEE NORCHIP conference, 14-15 Nov. 2011, Lund, Sweden.

[11] B. Widrow, I. Kollár, "Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications," Cambridge University Press, Cambridge, UK, 2008.