Approximating Communication Systems: Reality or Fantasy?

Ish Kool[†], Alish Kanani*[§], and Rajat Bhattacharjya*[†]
[†]Indian Institute of Information Technology Guwahati, Assam, India
[§]Indian Institute of Technology Jodhpur, Rajasthan, India
kool382@gmail.com, kanani.1@iitj.ac.in, rajatquiz@gmail.com

***Abstract*—The line of development towards 5G has let the world to rethink its hardware innovations. This is the time of data explosion and battery-operated ubiquitous devices are enormous in nature. Therefore, it has become ever important to speed up processing capabilities in edge devices for compute efficiency. But just increasing compute power doesn't work as we need to take care of the fact that edge devices are battery-operated. So it is also important to take care of power efficiency in edge devices. The domain of *Approximate Computing* offers a conclusive method to speed up compute and increase area and power efficiency in electronic devices for error-resilient applications.**

Communication systems involve signal processing blocks which perform DFT/IDFT. These signal processing algorithms are compute-intensive in nature, and research efforts have shown that they possess error-resilience due to their compute pattern which makes them a good candidate for approximations. In this paper, we investigate a technique so as to lower computational complexity in an FFT/IFFT which is extensively done inside a DSP. Our approach shows that we have a marginal loss in accuracy and on average our method gives 2.56%, 3.16% and 3.55% error in BER for 4, 16 and 64 point FFT respectively.

I. INTRODUCTION

The past decade brought a lot of improvements in wireless connectivity specifically the cellular communication saw improved spectrum efficiency and high data rate, all thanks to LTE [1]. Mobile broadband thus has become an important part of future cellular communication and the telecommunication market has witnessed opening of new opportunities and a sharp rise in growth [2].

LTE is a very capable technology which will see continuous evolution over time. The wireless technology is ever evolving and improving and hence to cater to the networks of future generation that will suit a wide range of use cases and an even wider range of specific requirements a new 5th generation radio access has been standardised. The New Radio (NR) improves on the data rate aspects, promises extremely low latency, delivers high reliability all while extending and making extremely efficient use of the most important and expensive resource the air spectrum [3].

One of the key challenges that 5G tech will face is increased power consumption due to increased antenna numbers, larger bandwidths to support and higher base station density [4]. Power consumption is an important factor when it comes to designing IoT devices and smartphones. A lot of thought and research has gone into the design of NR from this viewpoint for instance discontinuous reception, dynamic indication of control channel monitoring, dynamic antenna adaptation, etc. [5] and considerable amount of efforts are being leveraged to minimize power consumption wherever possible [4].

On the same lines, we propose a solution to counter this challenge and we do it by focusing on the very fundamental operation that all the NR and LTE devices will use.

Like LTE, the 5G NR uses OFDM as its key waveform for its air interface. OFDM solves a lot of problems that plagued communication systems of the past generations. It effectively counters selective fading, improves spectrum efficiency, reduces intersymbol interference (ISI) and inter-carrier interference (ICI) by clever use of cyclic prefix [6]. Despite these efficient properties, performing OFDM in transmission and reception is a computationally intensive task due to the fact that the rate of advancements in realizing performance intensive digital signal processors have not been able to keep up with the demands of advanced communication systems [7]. The DFT on reception and IDFT on transmission are highly computationally intensive operations a digital signal processor (DSP) preforms in a device while receiving or transmitting data [8].

Past research shows that these operations are error resilient in nature due to their compute pattern [9]. The error resilience possessed by algorithms such as DFT/IDFT can be attributed to factors like noise and redundancy in real-world data and the application's capability of mitigating accuracy loss as a result of efficient compute patterns. Thus our main investigation will be in finding to what extent these computationally intensive

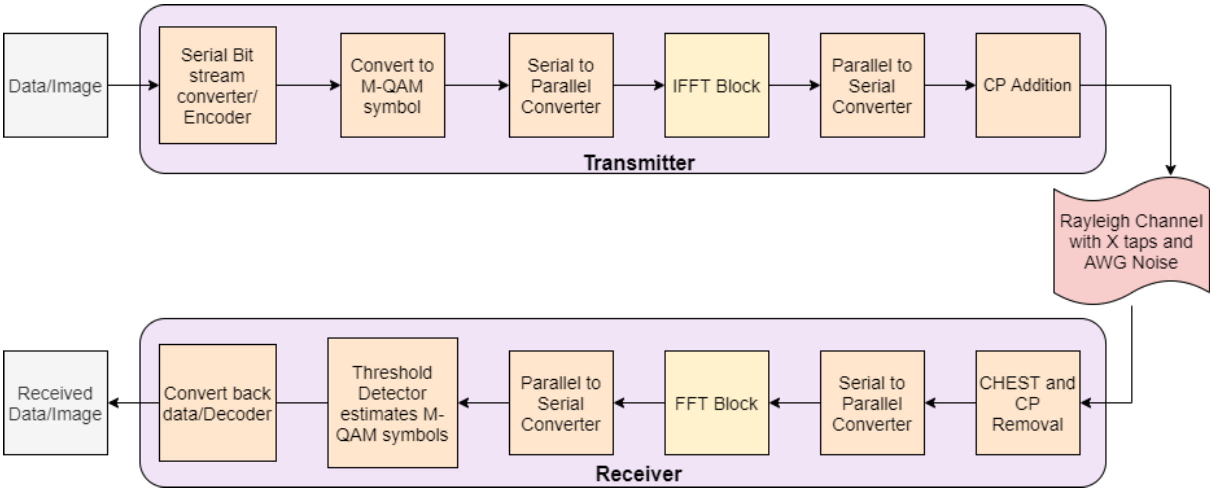

Fig. 1: Fundamental OFDM Communication System

tasks can be approximated and by doing so how much power consumption in a transceiver can be minimized.

Towards bridging the domain of approximate computing and communication systems, we have explored a methodology where we approximate the computationally intensive Fast Fourier Transform in a user equipment (UE). Our method involves the use of approximate arithmetic circuits inside a DSP which would reduce the computational complexity an FFT/IFFT has. We use an already existing library of approximate adders [10] for validating our claim.

II. BACKGROUND AND MOTIVATION

It has become ever important to ramp up speed and power efficiency in electronic devices. The era of 5G has brought up the scenario of ever-increasing data explosion. Such huge chunks of data need fast processing at the edge.

A. *Fundamental Communication Model*

Figure 1 shows the fundamental OFDM communication system with a transmitter and a receiver communicating over a Rayleigh channel with additive white Gaussian noise and Figure 2 represents the discrete time base-band model of the same communication system. The transmitter converts the data to be sent into a serial stream of bits, an advanced encoder can be used with a specific encoding algorithm to increase the redundancy and thus guarantee less error rate at the expense of decreased code rate but to keep the model simple we do not use any encoding scheme, which translates to code rate being unity. These bits are converted into M-QAM symbols where each symbol has complex signal representation. The value of M can be selected so that the modulation scheme is BPSK, QPSK, 16-QAM or 64-QAM. Equation 1 represents a block of symbols to be transmitted d where $d[0], d[1], ..., d[N-1]$ are M-QAM symbols.

$$\mathbf{d} = [d[0], d[1], ..., d[N-1]]^t \tag{1}$$

These symbols are fed to a serial to parallel converter which spreads it over the bandwidth by sending out N streams to the IDFT block to perform N-Point IDFT on them, the N here is the number of sub-carriers over which the actual transmission happens. From there parallel to serial converter gives out the time domain complex samples, each sample is now an OFDM symbol. Cyclic prefix is applied to each OFDM symbol to counter inter symbol interference (ISI) which may happen due to fading over the channel and then the samples are finally pushed over the air. Equation 2 represents the OFDM symbol after the application of cyclic prefix.

$$\begin{aligned} s = & [d[N-X+1], d[N-X+2], ..., \\ & d[N-1], d[0], d[1], ..., d[N-1]]^t \end{aligned} \tag{2}$$

The receiving antenna receives the transmitted signals from various paths which may be line of sight and non line of sight and as a result each path has its own attenuation and distortion factor also known as fading. There are numerous ways of modelling fading effects in the literature and each one of them has been studied in great depth, however for our use case which is cellular communication under dense environments, Rayleigh fading is a good statistical assumption for modelling

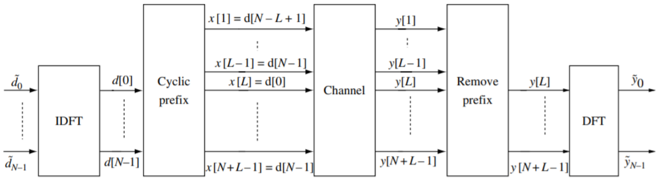

Fig. 2: Discrete time base-band model

multi-path receptions [11]. For simplicity we assume that the channel is underspread and hence approximately time-invariant for sufficiently long time-scale, in other words it means that the channel estimation can be done accurately with known signals [12]. Thus the discrete time channel model with finite X taps constant over time can be described as given in Equation 3.

$$y[m] = \sum_{l=0}^{l=X-1} h_l[m]s[m-l] + w[m] \tag{3}$$

Where hl[m] is the complex channel filter tap at time m, w[m] is the gaussian noise, s[m] is the sample to be transmitted and X=Td*W (symbol time W=bandwidth).

Thus the receiving antenna receives faded signals with Gaussian noise added to them. The channel estimator block performs channel estimation based on the received but already known signals also known as pilot signals to estimate the channel taps $\bar{\mathbf{h}}$. Cyclic prefix safeguards OFDM symbols which would have otherwise succumbed to ISI due to fading. Cyclic prefix is removed to obtain the OFDM symbol.

The OFDM symbols thus obtained are then fed to FFT block to perform N-point FFT to obtain the frequency domain complex samples. The output vector after FFT operation is given in Equation 4.

$$\bar{y}_n = \bar{h}_n \bar{d}_n + \bar{w}_n \; ; where \, n = 0, ..., N-1 \tag{4}$$

The threshold detector makes optimal decisions based on the energy level and determines the estimated M-QAM symbols $\bar{d}_n$.

III. METHOD

As stated before, the channel being underspread, the channel estimation becomes quite accurate and thus our model as shown in Figure 1 translates to Figure 3.

From Figure 3, we can see that we send BPSK data from the transmitter and there is IFFT performed on it. Over the channel there is addition of additive white gaussian noise (AWGN), and in the receiving end we see FFT being performed, followed by threshold detection to retrieve the BPSK data.

Figure 4 is a replication of Figure 3, but the FFT performed in Figure 4 is approximate in nature. In Figure 4, we approximate the addition operations involved in the FFT with the help of the approximate adders as listed in the EvoApprox Library [10]. The point of approximating FFT is from the fact that FFT can withstand some errors induced by approximations, given the efficient compute pattern of it [9].

We approximate the IFFT and FFT in the communication model.

An FFT can be given as:

Let m be an integer and let $N = 2^m$. Suppose that $x = [x_0, \ldots, x_{N-1}]$ is an N dimensional complex vector. Let $\omega = \exp(\frac{-2\pi i}{N})$. Then the DFT (Discrete Fourier Transform), $c = F_N(x)$ is given by

$$c_k = \frac{1}{N} \sum_{j=0}^{j=N-1} x_j \omega^{jk}. \tag{5}$$

Let $n = N/2$ and let u and v be n dimensional vectors defined by

$$u_j = x_j + x_{j+n}, \; j = 0, \ldots, n-1 \tag{6}$$
$$v_j = (x_j - x_{j+n})\omega^j, \; j = 0, \ldots, n-1. \tag{7}$$

Then

$$c_{2j} = \frac{1}{2}(F_n(u))_j, \; j = 0, \ldots, n-1 \tag{8}$$
$$c_{2j+1} = \frac{1}{2}(F_n(v))_j, \; j = 0, \ldots, n-1. \tag{9}$$

Though the major goal of using approximate computing is to speed up compute in addition to saving area and power, we must note that our final result should not deviate too much from the expected one as that would materialise to nothing ultimately.

We approximate the addition operations in FFT and IFFT as can be deduced from the summation shown in Equation 5 and its inverse.

The approximations incorporated in IFFT and FFT helps us speed up compute in addition to saving on chip area and power as can be seen from Section IV. This also doesn't impact the final result drastically and there is an extremely low deviation in the BER compared to what is observed when the IFFT and FFT are accurate in nature.

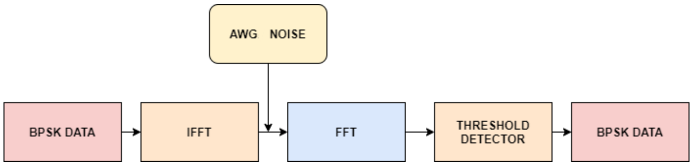

Fig. 3: A Communication System with BPSK

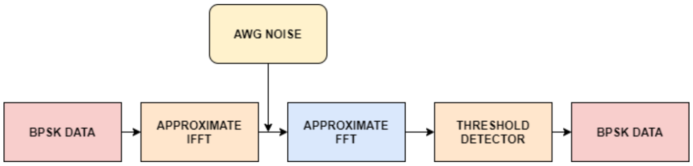

Fig. 4: An Approximate Communication System with BPSK

IV. RESULTS AND ANALYSIS

To evaluate the approximation in OFDM communication, we first perform the functional validation, i.e, the accuracy analysis after approximating additions at the software level. Then to check hardware efficiency, we will explain pareto optimal method used to create EvoApprox library [10]. The main motivation of using a library is the future scope and scalability as any new approximate adder with better performance can directly be incorporated into approximate OFDM communication system.

A. Functional Validation

At the software side for functional validation, we model the systems as shown in Figure 3 using python and replaced all the accurate add operation in FFT and IFFT blocks with approximate adders from EvoApprox 16-bit signed adder library. Figure 4 highlights approximate FFT and IFFT.

For accuracy analysis and adder selection, we calculate the bit error rates (BER) for accurate and approximate system at SNR 6dB. For experiment purpose we have used 4,16 and 64 point FFT and IFFT and compare them for various types of adders as shown in Table I.

After incorporating all the Pareto optimal approximate 16-bit sign-extended adders available in the EvoApprox library, we have concluded that BER does not variate much in comparison to accurate system. From table I we can say for 4,16 and 64 point FFT/IFFT, approximate system gives on an average 2.56%, 3.16% and 3.55% error in BER for above mentioned operating condition. We have observed that add16se_2AS and add16se_2E1 gives more error in BER compare to others but it will have low power consumption.

TABLE I: Bit Error Rates using different types of adders

Adder Type	**4**	**16**	**64**
Accurate	2.39500E-03	2.39750E-03	2.37094E-03
add16se_1Y7	2.39000E-03	2.39250E-03	2.39250E-03
add16se_20J	2.38500E-03	2.38500E-03	2.37531E-03
add16se_25S	2.31500E-03	2.36125E-03	2.40406E-03
add16se_26Q	2.39000E-03	2.39500E-03	2.37063E-03
add16se_28H	2.39500E-03	2.42500E-03	2.42969E-03
add16se_294	2.37500E-03	2.41500E-03	2.37969E-03
add16se_29A	2.50500E-03	2.56125E-03	2.51406E-03
add16se_2AS	2.87500E-03	2.93625E-03	3.01563E-03
add16se_2BY	2.37500E-03	2.38125E-03	2.35344E-03
add16se_2DN	2.38500E-03	2.39375E-03	2.37094E-03
add16se_2E1	2.59000E-03	2.70250E-03	2.72469E-03
add16se_2GE	2.33500E-03	2.36250E-03	2.34094E-03
add16se_2H0	2.39500E-03	2.39875E-03	2.37188E-03
add16se_2JB	2.37500E-03	2.49000E-03	2.48063E-03
add16se_2JY	2.38500E-03	2.38125E-03	2.36781E-03
add16se_2KV	2.38500E-03	2.38375E-03	2.37250E-03
add16se_2LJ	2.38500E-03	2.39875E-03	2.36875E-03

B. Adder Characteristics

The adders listed in Table I have an optimum balance between their inherent accuracy and hardware characteristics as they are genetically programmed. Their characteristics can be well understood from Figure 5 which shows pareto optimal adders. For this task [13] used Power vs Error Probability or Error Rate (EP), Mean Absolute Error (MAE), Worst-Case absolute Error (WCE) and Mean Relative Error (MRE) graphs and choose the optimal adders. As can be explained by Figure 5 add16se_2AS and add16se_2E1 adders have low power consumption but at the same time high error metric at the same time, so based on the accuracy requirement and Power-Performance-Area(PPA) requirement of the application, one can choose any adder from this list.

V. RELATED WORKS

In the last decade, there have been a lot of works done in the domain of approximate computing [14]. Starting from the designs of new approximate arithmetic circuits [15]–[18] to applications such as low power digital signal processing using approximate adders with specific quality constraints [19], DNN acceleration using approximate computing [20] etc. have been proposed extensively in existing literature.

The domain of communication systems has remained untouched compared to other domains with regard to approximate computing. Hence, in this paper we have investigated a method to approximate operations in a communication system so as to gain hardware efficiency.

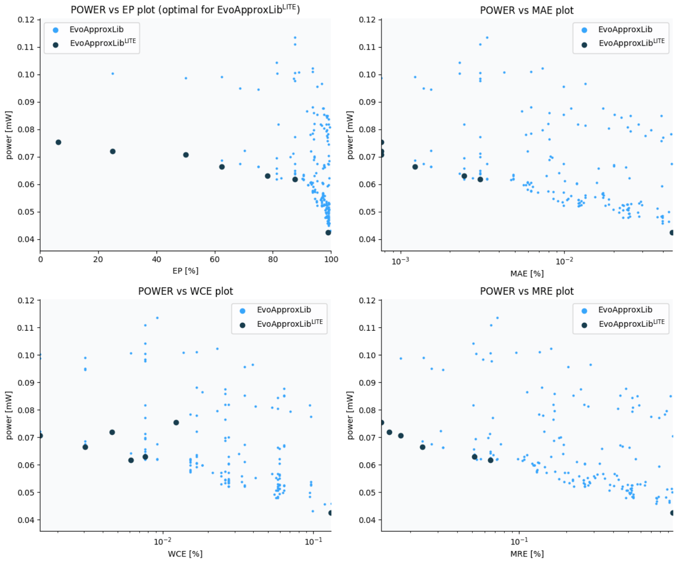

Fig. 5: EvoApprox Adders' Characteristics (16-bit sign extended) [10]

VI. CONCLUSION AND FUTURE WORK

From the results obtained we conclude that approximating FFT/IFFT blocks in a transceiver yields very low and tolerable error margins, also using the hardware adders and multipliers will make the overall PPA better due to their inherent power efficient nature.

In this paper, we have simulated 4,16 and 64 point approximate FFT/IFFT algorithms. To further evaluate our claim of error resilience we plan to extend the approximate designs for FFT/IFFT blocks to higher sizes and synthesize the hardware block for accurate PPA and perform simulations over a good range of SNR values and dynamic channel conditions. Thus, it answers the basic question this paper asks, i.e., can we approximate communication systems? Yes, we can.

REFERENCES

[1] M. Marques da Silva and F. Monteiro, *MIMO Processing for 4G and Beyond: Fundamentals and Evolution*, 06 2014.

[2] I. P. Chochliouros, A. Kostopoulos, A. S. Spiliopoulou, A. Dardamanis, I. Neokosmidis, T. Rokkas, and L. Goratti, "Business and market perspectives in 5g networks," in *2017 Internet of Things Business Models, Users, and Networks*, 2017, pp. 1–6.

[3] M. H. Alsharif and R. Nordin, "Evolution towards fifth generation (5g) wireless networks: Current trends and challenges in the deployment of millimetre wave, massive mimo, and small cells," *Telecommunication Systems*, vol. 64, no. 4, pp. 617–637, 2017.

[4] C.-L. I., S. Han, and S. Bian, "Energy-efficient 5g for a greener future," *Nature Electronics*, vol. 3, pp. 182–184, 04 2020.

[5] T. Kim, Y. Kim, Q. Lin, F. Sun, J. Fu, Y. Kim, A. Papasakellariou, H. Ji, and J. Lee, "Evolution of power saving technologies for 5g new radio," *IEEE Access*, vol. 8, pp. 198 912–198 924, 2020.

[6] A. A. Zaidi, R. Baldemair, H. Tullberg, H. Bjorkegren, L. Sundstrom, J. Medbo, C. Kilinc, and I. Da Silva, "Waveform and numerology to support 5g services and requirements," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 90–98, 2016.

[7] C. She, P. Cheng, A. Li, and Y. Li, "Grand challenges in signal processing for communications," *Frontiers in Signal Processing*, vol. 1, p. 2, 2021. [Online]. Available: https://www.frontiersin.org/article/10.3389/frsip.2021.664331

[8] K. Elango and M. Kannan, "A survey on fft/ifft processors for next generation telecommunication systems," *Journal of Circuits, Systems and Computers*, vol. 27, 03 2018.

[9] B. S. Prabakaran, S. Rehman, and M. Shafique, "Xbiosip: A methodology for approximate bio-signal processing at the edge," ser. DAC '19. New York, NY, USA: Association for Computing Machinery, 2019.

[10] V. Mrazek, L. Sekanina, and Z. Vasicek, "Libraries of approximate circuits: Automated design and application in cnn accelerators," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 406–418, 2020.

[11] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .i. characterization," *IEEE Communications Magazine*, vol. 35, no. 7, pp. 90–100, 1997.

[12] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[13] P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, "Approximate Multiplier Design Using Novel Dual-Stage 4:2 Compressors," *IEEE Access*, vol. 8, pp. 48 337–48 351, 2020.

[14] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *2013 18th IEEE European Test Symposium (ETS)*, 2013, pp. 1–6.

[15] S. Belwal, R. Bhattacharjya, K. Goswami, and D. S. Banerjee, "Acla: An approximate carry-lookahead adder with intelligent carry judgement and correction," in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*, 2021, pp. 1–7.

[16] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "RAP-CLA: A Reconfigurable Approximate Carry Look-Ahead Adder," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 8, pp. 1089–1093, 2018.

[17] R. Bhattacharjya, A. Kanani, and N. Goel, "Rearm: A reconfigurable approximate rounding-based multiplier for image processing," in *2020 24th International Symposium on VLSI Design and Test (VDAT)*, 2020, pp. 1–4.

[18] S. Vahdat, M. Kamal, A. Afzali-Kusha, M. Pedram, and Z. Navabi, "Truncapp: A truncation-based approximate divider for energy efficient dsp applications," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*, 2017, pp. 1635–1638.

[19] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, 2013.

[20] V. Mrazek, Z. Vasicek, L. Sekanina, M. A. Hanif, and M. Shafique, "Alwann: Automatic layer-wise approximation of deep neural network accelerators without retraining," *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2019. [Online]. Available: http://dx.doi.org/10.1109/ICCAD45719.2019.8942068