

# FPGA Implementation of Efficient 2D-FFT Beamforming for On-Board Processing in Satellites

Rakesh Palisetty<sup>\*</sup>, Geoffrey Eappen<sup>\*</sup>, Vibhum Singh<sup>\*</sup>, Luis Manuel Garces Socarras<sup>\*</sup>, Vu Nguyen Ha<sup>\*</sup>, Juan A. Vásquez-Peralvo<sup>\*</sup>, Jorge Luis Gonzalez Rios<sup>\*</sup>, Juan Carlos Merlano Duncan<sup>\*</sup>, Wallace Alves Martins<sup>\*</sup>, Symeon Chatzinotas<sup>\*</sup>, Björn Ottersten<sup>\*</sup>, Adem Coskun<sup>†</sup>, Stephen King<sup>†</sup>, Salvatore D’Addio<sup>†</sup>, and Piero Angeletti<sup>†</sup>

<sup>\*</sup>Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Luxembourg

<sup>†</sup>European Space Agency

Emails: (rakesh.palisetty, geoffrey.eappen, vibhum.singh, luis.garces, vu-nguyen.ha, juan.vasquez, jorge.gonzalez, juan.duncan, wallace.alvesmartins, symeon.chatzinotas, bjorn.ottersten)@uni.lu, (adem.coskun, salvatore.daddio, piero.angeletti)@esa.int, stephen.king@ext.esa.int

**Abstract**—On-board processing of digital beamforming in satellites is an efficient solution for the higher data rates, more capacity, and lower latency, but the available on-board limited power makes it impractical to digitally create thousands of beams at once. A significant portion of the analog hardware in a satellite communications payload can be replaced with highly integrated digital components, which are often more affordable, lighter, smaller, and reprogrammable by employing digital beamforming. In comparison to matrix-by-vector multiplication beamforming, the discrete Fourier transform (DFT) beamformer enables the finer realization of real-time beamformers with reduced circuit complexity and lower power consumption. Fast Fourier transform (FFT) methods can further reduce the computing cost of the DFT computation. Therefore, in this paper, area-power efficient two-dimensional (2D) FFT digital beamforming techniques are analyzed and implemented. The major implementation challenge is to produce  $N$  samples per cycle with lower area-power consumption. Fully unrolled 4-bit twiddle factor (TF) quantized FFT is proposed in this regard. The optimization techniques through quantization, truncation, and complex multipliers are thoroughly discussed for efficient implementation. The behavioral and post-route timing simulations are validated, and implementation results like area and power consumption are estimated and compared among conventional, fully unrolled, and the proposed 4-bit TF quantized 2D-FFT.

**Index Terms**—beamforming, fast Fourier transform, look-up table, power estimation, quantization.

## I. INTRODUCTION

The increasing number of users of satellite-based services is pushing the requirements for high speeds, increased performance, and more capacity [1]. In order to fulfill these requirements, satellites must be equipped to generate thousands of beams across the coverage region, necessitating a substantial capacity for aggregated beamforming. Analog beamformers suffer from lack of flexibility and high consumption of power and mass [2]. Feeder and reflector schemes stand as the practical alternative to achieve desired gains. When it comes to implementing direct radiating arrays (DRA), hybrid beamforming techniques represent the sole available option [3]. However, the implementation of hybrid beamforming carries with it a trade-off between complexity and flexibility [4]. The advancements in digital hardware-based design have rapidly lowered the cost

and increased the abilities of digital components employed in digital beamformers. In this context, on-board processors can deal with a limited portion of the system’s capacity with respect to digital beamforming. Reducing power consumption would enable the deployment of a fully digital payload, enhancing capacity allocation flexibility to cater to a wider range of user applications [5]. These requirements can be translated into an area and power-efficient digital beamformer without impairing the flexibility of fully digital solutions.

Digital beamforming through matrix-by-vector multiplication operation is a brute-force method, and each scalar multiplication is a hardware process that consumes a lot of space and power. When employing uniformly linear or rectangular arrays, the apparent choice for implementing codebook-based digital beamforming is to utilize the discrete Fourier transform (DFT) as the beamforming matrix [6]. The fast Fourier transform (FFT) algorithm can therefore be used to implement the DFT effectively [7]. By leveraging FFT methods, the computing cost of the DFT computation can be reduced to  $\mathcal{O}(N \log N)$ . This reduction in complexity elucidates why real-time beamformers can be implemented more efficiently, requiring fewer circuitry and power compared to matrix-by-vector multiplication.

Implementation of two-dimensional (2D) FFT digital beamforming requires  $N$  output samples per clock cycle. The conventional FFT in [8] takes  $N$  clock cycles for  $N$  samples, which does not fulfill the requirement to perform FFT-based digital beamforming. A fully unrolled FFT is capable of producing  $N$  samples per clock cycle [9]. As mentioned in [10], the performance of the fully unrolled FFT-based digital beamforming on satellite systems can be effectively improved in terms of power reduction, area reduction, and increased throughput. While efficient FFT algorithms enable the realization of completely unrolled FFT beamforming, this approach might still be overly complex for certain satellite applications. In such cases, it becomes crucial to optimize the utilization of on-board resources like power consumption and mass. The twiddle factors (TF) consume more power resources due to the existence of the multiplication process.

Quantization of TFs can be proven to be an efficient solution in fully unrolled FFT architectures for digital beamforming applications.

Therefore, in this paper, we have proposed 4-bit TF quantized fully unrolled 2D-FFT digital beamforming for on-board processing in satellites. First, we examine the conventional FFT complexity, which operates in a rolled fashion, processing one input sample per cycle of operation. Subsequently, we delve into the fully unrolled FFT, which processes  $N$  input samples per clock cycle of operation, eliminating the rolled process. The implementation of a fully unrolled FFT results in higher area and power consumption. To address this concern and achieve further reduction in both area and power usage, a 4-bit TF quantized FFT is proposed for the 2D-FFT beamforming design. The proposed 4-bit TF quantized fully unrolled FFT is analyzed with signal-to-noise ratio (SNR) measurement and usage of truncation/rounding mode in implementation. Furthermore, the detailed 2D-FFT implementation methodology with the optimization techniques and pipelining strategy is presented for the proposed architecture. The 2D-beamforming plot with the proposed 4-bit TF quantized FFT is discussed, and the implementation results of the 2D-FFT beamforming in a typical medium Earth orbit (MEO) satellite scenario.

The remaining sections of the paper are structured as follows. In Section II, we present an analysis of FFT architectures, covering aspects such as computational complexity, effects of quantization, and SNR measurement. Then, the implementation methodology of 2D-FFT in unrolled fashion for 4-bit TF quantized FFT is presented in Section III. Lastly, we discuss a preliminary evaluation of the 2D-FFT simulations in MATLAB and field programmable gate arrays (FPGA) concerning the implemented area-power consumption for the FFT-based onboard digital beamforming in various scenarios in Section IV. Subsequently, we offer concluding remarks in Section V.

## II. ANALYSIS OF FFT ARCHITECTURES

The conventional DFT entails  $N \times N$  multiplications and  $(N - 1) \times N$  additions, resulting in significant computational complexity. As a more efficient alternative, FFT structures are employed in beamforming, reducing the multiplication complexity to  $N/2 \times \log_2(N)$  and addition complexity to  $N \times \log_2(N)$ . Radix-4 based FFT requires 25% fewer multipliers when compared to radix-2 even though the area of adders remains the same in both cases [10]. In radix-4, the theoretical total number of multipliers is equivalent to  $\frac{3N}{4} \times \log_4(N)$ , while the number of adders is  $N \times \log_2(N)$ .

The derived generalized equation for obtaining the optimized complex multipliers and complex adders for different FFT sizes using radix-4 can be expressed as follows:

- Number of complex multipliers =  $\sum_{n=0}^{m-1} \left( \frac{3N}{4} - 3 \times 4^n \right)$   
where  $m = \log_4(N/4)$  for FFT size  $\geq 16$
- Number of complex adders =  $N \times \log_2(N)$

One input sample is processed in a rolled fashion by the FFT in commercially accessible products like FPGA. If  $N$  samples are required at once, which is the case for beamforming, then utilizing this type of architecture will need multiplying the hardware or frequency  $N$  times. A completely unrolled FFT is capable of producing  $N$  samples at once [9]. Furthermore, in the context of a fully unrolled architecture, the estimated number of multipliers is lower than the theoretical computation. Additionally, this fully unrolled approach offers resource reduction when not all FFT outputs are required for the digital and radio-frequency (RF) chains. When designing the multiplier for implementation, the twiddle factors (TFs)  $W_0^N = 1$  are excluded since they only result in data multiplied by one. Despite the attractiveness of fully unrolled FFT for beamforming operations, it comes with a substantial demand for area and power consumption. As a result, a 4-bit TF quantized FFT is proposed to mitigate these requirements. In the subsequent subsections, it becomes evident that the 4-bit TF quantized FFT maintains linear operation without introducing interference.

### A. FFT with TF Quantization

The critical component in deciding the number of multiplier operations and the number of look-up tables (LUTs) occupied by the multiplier is TFs. In this subsection, we analyze the FFT with different quantization levels so that the quantized FFT gives a similar SNR performance as that of conventional FFT. In the extreme case, when all the TFs in the radix-4 FFT are rounded to the unity, the ‘‘DFT-like’’ transformation is not exactly a DFT but is exactly a complex Hadamard transformation. On the other hand, the advantage of quantized FFT will have a smaller number of LUTs occupied compared to conventional FFT. The plots in Fig. 1a, Fig. 1b, and Fig. 1c present the real-part of TF for the last stage of a FFT quantized with 4-bit, 6-bit, and 8-bit, respectively. From the plots in Fig. 1a, Fig. 1b, and Fig. 1c, it is observed that the TF quantized FFT has a similar result compared to conventional FFT. Furthermore, the equivalent SNR measured from mean squared error for a complex Gaussian random input with respect to the conventional FFT is presented in Fig. 2a, Fig. 2b, and Fig. 2c. From Fig. 2a, Fig. 2b, and Fig. 2c, it is observed that the SNR is 24.6 dB, 35.6 dB, and 47.9 dB respectively, and approximately equal to the theoretical SNR given by  $\text{SNR} \approx 6b + 1.72$ , with  $b$  being the number of bits.

### B. Truncation and Rounding in FFT

In this subsection, the analysis of employing truncation instead of rounding in the implementation is presented. The simulations using both rounding and truncation are presented in Fig. 3a and Fig. 3b. The  $x$ -axis represents the repetition number of FFT operations and  $y$ -axis denotes the SNR in dB. The input random Gaussian data samples to the FFT are in the format of Q(16,15) i.e., one bit for sign and fifteen bits for the fractional part. In Fig. 3a, the input samples provided to the FFT are the random Gaussian samples, and it is observed that the rounding technique has an SNR of 69.31 dB and

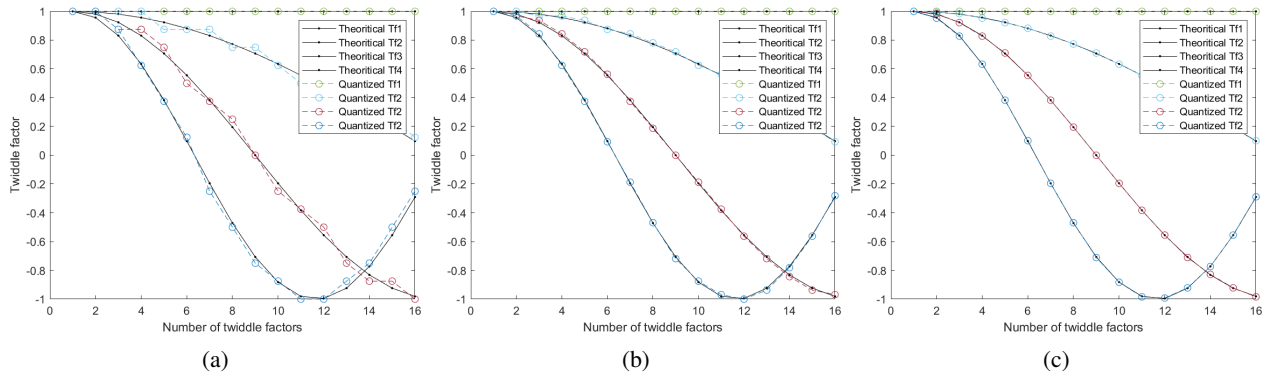


Fig. 1: Comparison of theoretical TFs with (a) 4-bit, (b) 6-bit, (c) 8-bit, TF quantization.

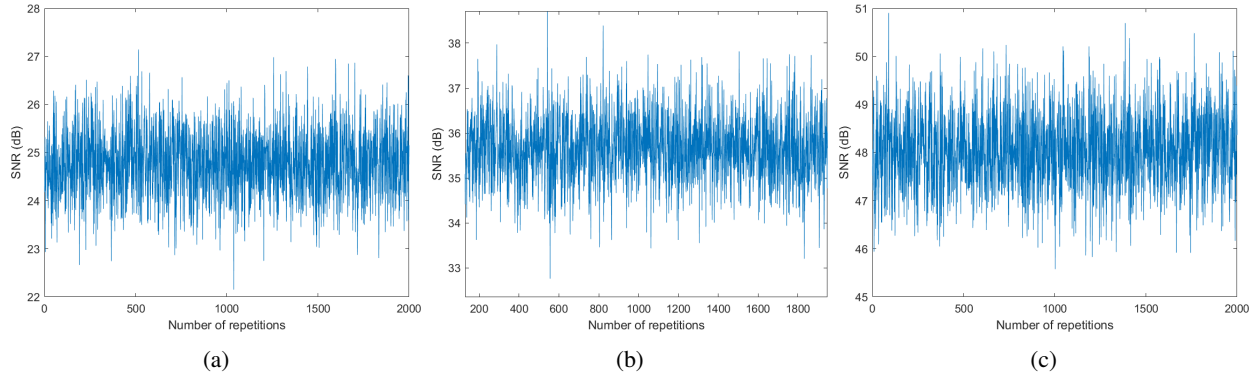


Fig. 2: SNR measured for conventional FFT with (a) 4-bit, (b) 6-bit, (c) 8-bit, TF quantization.

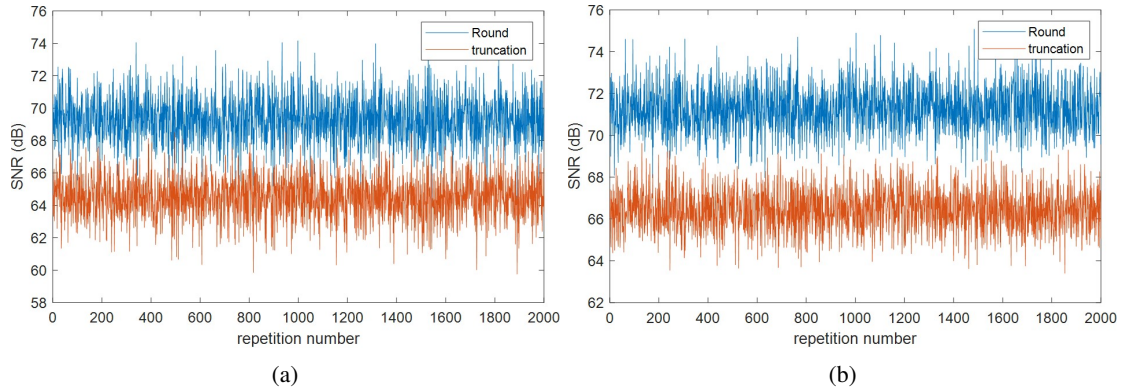


Fig. 3: Rounding and truncation for 16-point FFT with (a) random Gaussian samples, (b) random Gaussian samples and sinusoidal signal.

truncation has an SNR of 64.55 dB. The overall difference in SNR between them is around 4.75 dB which indicates that there is a precision loss of roughly around 1 bit by employing truncation in the implementation i.e.,  $\text{SNR} = 6.02b + 1.72$  dB. On the other hand, the advantage of using the truncation helps in reducing the number of extra additions required for employing the rounding technique i.e., by  $8(N/4) \times \log_4(N)$ .

Furthermore, for a better understanding of this performance, a random Gaussian sample added with a sinusoidal signal is provided as an input, and the obtained results are presented in

Fig. 3b. It is noted that rounding is 71.26 dB and truncation is 66.45 dB. The overall difference in SNR accounts for 4.81 dB which signifies a loss of precision by 1-bit by employing truncation. So, we can say that even though truncation leads to a loss of precision by 1-bit, we were able to save the number of additions required for performing the quantization operation by a factor of  $8(N/4) \times \log_4(N)$ . Based on the advantage of fully unrolled FFT architectures, the 4-bit quantization performance with its SNR measurement, the detailed implementation with optimization of proposed fully unrolled 4-bit TF quantized FFT is discussed in Section III.

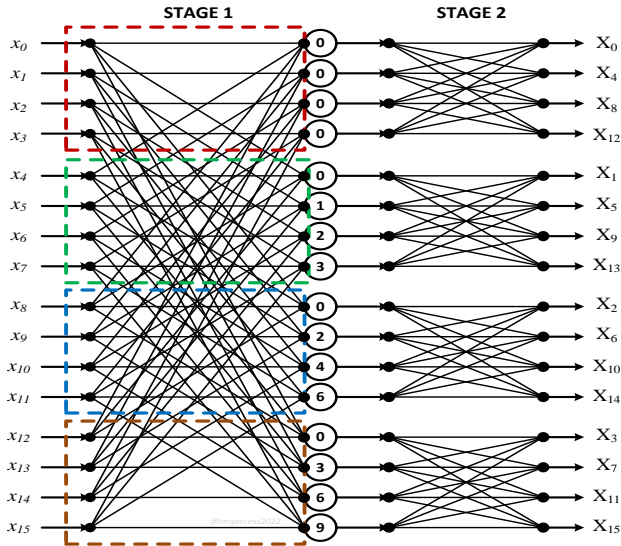


Fig. 4: Architecture of fully unrolled FFT.

### III. IMPLEMENTATION METHODOLOGY

The fully unrolled architecture of 16-point FFT employed for the implementation is presented in Fig. 4, and the same architecture is used for 4-bit TF quantized FFT. In the fully unrolled FFT architecture, the TF has a bit-width of sixteen bits, whereas, in the proposed 4-bit TF quantized FFT, TF has a bit-width of four bits. The architecture in Fig. 4 employs a radix-4 algorithm that has two stages with four butterflies in each stage. Each stage consists of four butterflies, and eight butterfly modules are presented in total. The numbers circled after STAGE 1, i.e., 0000, 0123, 0246, and 0369, represent the TFs  $W_N^{nk}$ . Further, the red dotted module in Fig. 4 represents Butterfly 1 of STAGE 1 with no twiddle multiplications. The green color represents butterfly architecture's output multiplied by three TFs since one TF is zero ( $W_N^0 = 1$ ). Similarly, the remaining two butterflies have three TFs, and are highlighted in blue and brown colors. Since the output sample bit positions are shuffled after STAGE 2, the output sample bit positions are hardwired during implementation to represent normal order.

The intent in this implementation is to design an efficient complex multiplier with less power and area utilization. For understanding the implementation of the complex multiplication, the complex multiplier module implemented using three real multipliers is better understood with an example. The output of Butterfly 2 (green color) in STAGE 1 (let us say  $X_r + jX_i$ ) is multiplied with TF  $W_{16}^{1,1}$ , i.e.,  $e^{-j\frac{2\pi}{16} \cdot 1 \cdot 1}$  which is equivalent to  $0.9239 - j0.3827$ . We can realize them with real and imaginary parts of the sample  $X_r + jX_i$  multiplied with the above TF by considering temporary variables  $Z$ ,  $D$ , and  $E$ , which can be expressed as

$$\begin{aligned} Z &= 0.9239 \cdot (X_r - X_i), \\ D &= 0.9239 + 0.3827j, \\ E &= 0.9239 - 0.3827j. \end{aligned}$$

Thus, by performing  $D \cdot (X_i + Z)$  and  $E \cdot (X_r - Z)$ , the real and imaginary parts of the output of the multiplier are obtained. Hence, the complex twiddle multiplication is realized with three real multipliers and five real adders instead of four real multipliers and two real adders. Similarly, if the sample  $X_r + jX_i$  has to be multiplied with TF say  $0.7071 - j0.7071$ , then it is noted that ' $E$ ' is made zero, and hence one more real multiplier is reduced. Further optimizations include when TF of STAGE 1 Butterfly 3 say  $W_{16}^{2,2}$  i.e.,  $e^{-j\frac{2\pi}{16} \cdot 2 \cdot 2}$  (which is equivalent to  $0.0000 - j1.0000$ ) multiplied with  $X_r + jX_i$ . The resultant is equal to swapping the real and imaginary parts, with the imaginary being two's complements of the real sample, i.e., the real part is  $X_i$ , and the imaginary part is  $X_r + 1$ . Hence, no real multiplier is required in this case. Considering all these optimization techniques, an efficient, fully unrolled radix-4 FFT is implemented.

The following subsections signify the incorporation of pipelining technique (to achieve a higher operating frequency), and the construction of 2D-FFT based on the 1D-FFT discussed earlier.

#### A. Implementation of Fully Unrolled and 4-bit TF Quantized FFT with Pipelining

From the previous implementation architecture of the 16-point fully unrolled architecture in Fig. 4, the maximum operating frequency achieved is 83.3 MHz with a positive slack of 0.011 ns. This low operating frequency is due to the extensive combinational circuit path created due to the multiplier module, as shown in Fig. 5a. Similarly, the implemented design with 4-bit TF quantized FFT reported a maximum operating frequency of 125 MHz with a positive slack of 0.163 ns. Therefore, to increase the maximum operating frequency of the design, pipelining is introduced in the multiplier module. The employment of pipelining will improve the maximum operating frequency at the expense of increasing the number of flip-flops (FFs)/registers, as shown in Fig. 5b. The fully unrolled 16-point FFT with pipelining multiplier module achieved a maximum operating frequency of 129.534 MHz with a positive slack of 0.029 ns. Due to the insertion of FFs, the implemented design has an initial latency of six clock cycles, although the iteration interval is one clock cycle. Similarly, by employing the multiplier module with pipelining in the proposed fully unrolled 4-bit TF quantized 16-point FFT, an operating frequency of 212.766 MHz with a positive slack of 0.174 ns is achieved.

#### B. Implementation of 2D-FFT Beamforming

Implementation of 2D beamforming employs 2D-FFT operation. A 2D-FFT is performed comprising a row-wise operation and then followed by a column-wise operation. In this paper, a  $16 \times 16$  2D-FFT digital beamforming is performed. Thus, a total of sixteen FFTs of 16-points are required to accomplish the row-wise operation, and then sixteen FFTs of 16-points are required for column-wise operation, as shown in Fig. 6. For the fully unrolled architecture TFs are 16-bits in width and in the proposed 4-bit TF quantized 2D-FFT,

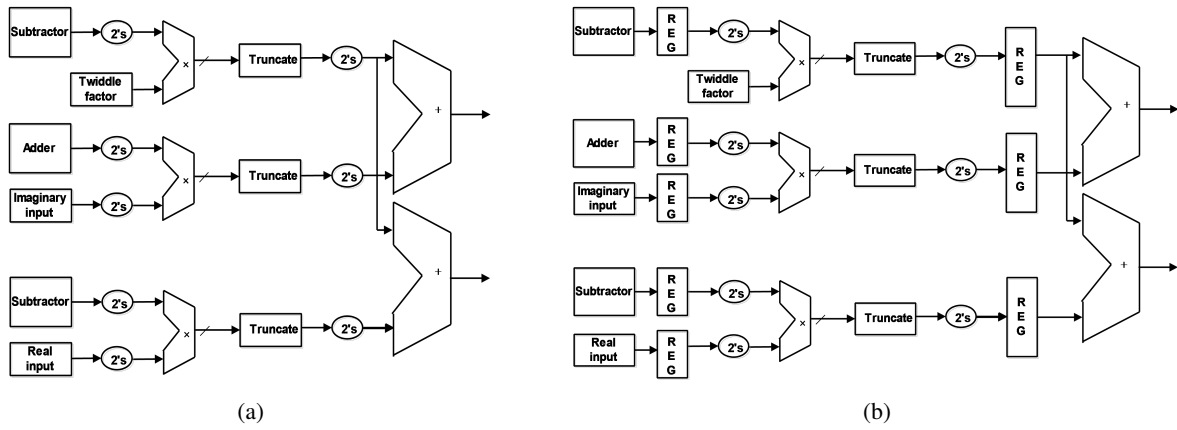


Fig. 5: Multiplier module for TF multiplication (a) without pipelining, (b) with pipelining.

each TF has 4-bits in width. It can be visualized that the two hundred and fifty-six inputs are fed to the sixteen blocks of 16-Point FFT-1 to perform row-wise FFT operation. Then, the rewiring block performs the re-connecting of the outputs from the sixteen blocks of 16-Point FFT-1 to the sixteen blocks of 16-Point FFT-2 for the column-wise operation. Inputs to the first block of the 16-point FFT-2 are denoted by the red wiring. Similarly, the blue wiring represents inputs to the second block

These weight values are obtained by performing 2D-FFT. The simulation plot in Fig. 7 shows the beam directivity in dB at  $0^\circ$  azimuth and elevation angles. The employed antenna dimension is  $16 \times 16$  and the azimuth and the elevation cut are  $-90^\circ$  to  $90^\circ$ . With different indexing to the input fed to the 2D-FFT, different beam directions are obtained as shown in Fig. 8, and it corresponds to different user locations. Here, indexing refers to the (row, column) position of the directed beams. Therefore, with 2D-FFT efficient beamsteering is possible in different directions with less complexity.

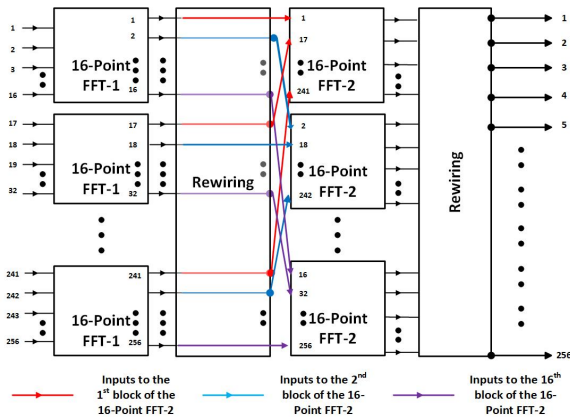


Fig. 6: Functional block diagram of 2D-FFT.

of the 16-point FFT-2. This process continues till the sixteenth block of the 16-point FFT-2, denoted by the purple wiring as shown in Fig. 6. Post FFT computation from the sixteen blocks of the 16-point FFT-2 the second stage of rewiring is performed to align the outputs.

#### IV. RESULTS AND DISCUSSIONS

This section presents the MATLAB simulations and the implementation results targeting xcvu29p-12fsga2577e FPGA.

##### A. 2D-FFT Beamforming Simulation Analysis

The MATLAB simulation for beamforming via 2D-FFT TF quantized is shown in Fig. 7 and Fig. 8. The 2D-FFT beamforming is carried out by multiplying the baseband signal with the weight values corresponding to the beamforming vectors.

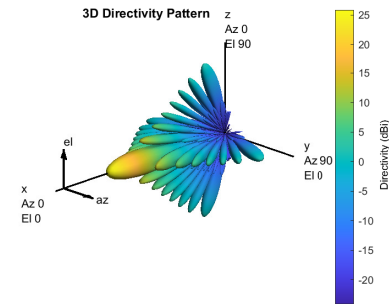


Fig. 7: 3D directivity beam pattern with indexing (1,1).

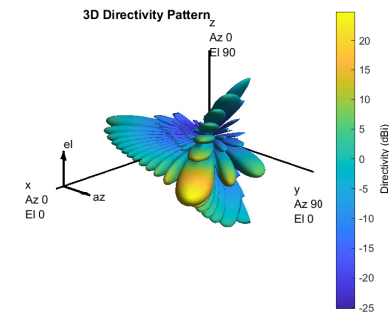


Fig. 8: 3D directivity beam pattern with indexing (1,12).

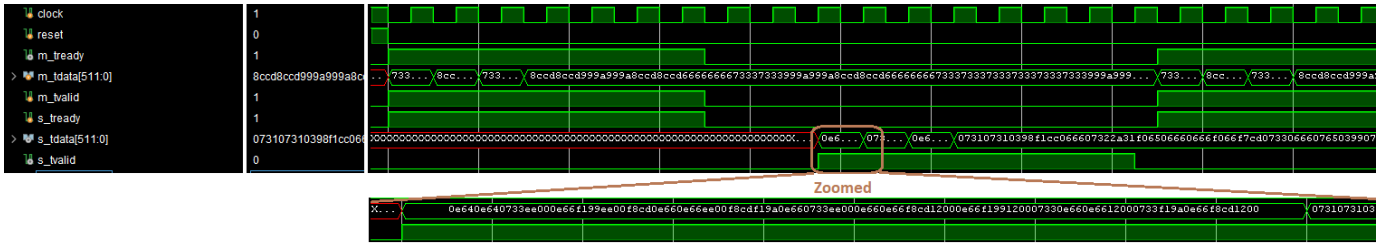


Fig. 9: Behavioural simulation of fully unrolled 4-bit TF quantized 1D-FFT.

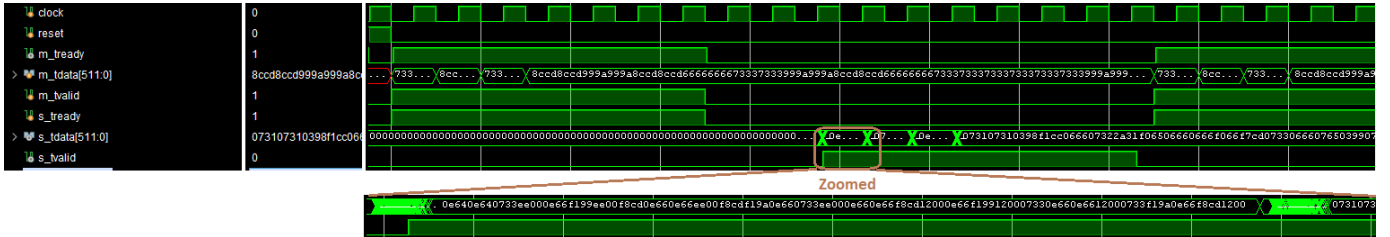


Fig. 10: Post-route timing simulation of fully unrolled 4-bit TF quantized 1D-FFT.

### B. FPGA Implementation Results

The implemented 2D-FFT has a structure of thirty-two 16-point FFTs with 256 inputs and 256 outputs. Since implementing a design with these numbers of high input/outputs (I/Os) is not feasible on a single FPGA due to I/O constraints, the out-of-context (OOC) synthesis was employed to estimate the area power consumption. Considering this, we have validated the proposed fully unrolled TF quantized 1D-FFT of 16-point with behavioral simulation and post-route timing simulation as shown in Fig. 9 and Fig. 10. Here axi4-stream protocol was incorporated whereby by the input data stream ‘m\_tdata’ has an input width of 512 bits (16 samples  $\times$  16 bits  $\times$  2 for real and imaginary ). The ‘m\_tvalid’ and ‘m\_tREADY’ indicate the valid and ready ports for input side. Similarly the output data ‘s\_tdata’ has corresponding 512 bits of data with valid and ready ports. The design runs on active low reset, and it can be notified from the Fig. 9 and Fig. 10 the outputs (which are zoomed) are generated per clock cycle, and are same in the both the cases.

The xcvu29p-12fsga2577e ultrascale+ FPGA is considered for implementing the proposed architecture with a operating frequency of 125 MHz (since the proposed 2D-FFT can be clocked at 230 MHz maximum, we considered 125 MHz frequency as the best scenario for extrapolation for 1500 MHz ). The power consumption and area utilization in terms of LUT, FFs, digital signal processing (DSP) blocks for conventional 2D-FFT, fully unrolled 2D-FFT, and the proposed 4-bit TF quantized 2D-FFT is presented in Table I. There were challenges when implementing the 2D-FFT using a conventional FFT algorithm since it is serial in nature and produces one sample per clock cycle. In this regard, there were two solutions to implement it. In order to obtain the required frequency of 125 MHz with each beam, the conventional 2D-FFT should be operated at 2 GHz (16 times the required frequency) using the architecture shown in Fig. 6. The second solution is to

implement a  $16 \times 16$  2D-FFT using  $256 \times 256$  FFT architecture since 256 output samples per clock cycle are needed at once. The second solution of conventional 2D-FFT was adopted while implementing the design since the first solution is not feasible to implement at 2 GHz on the FPGA.

From Table I, it is noticed that conventional 2D-FFT consumes 14.973 Watts (W) of dynamic power which is quite high compared to fully unrolled and 4-bit TF quantized 2D-FFT. The proposed 4-bit TF quantized 2D-FFT has a lower power consumption compared with the remaining two, and the main advantage is that there are no DSPs that can be used for other signal processing blocks such as sparse matrix (used for user selection) in the same FPGA.

TABLE I: Resource estimation for  $16 \times 16$  2D-FFT

Resources	Dynamic power consumption (W)	LUT	FF	DSP
Conventional 2D-FFT	14.973	563680	1372544	6144
Fully unrolled 2D-FFT	5.472	130208	112707	640
4-bit TF quantized 2D-FFT	5.419	142592	112643	0

TABLE II: Extrapolated resource estimation in MEO scenario

MEO mission reference scenario					
Resources	RF Chains	Power consumption (W)	LUT	FF	DSP
Conventional 2D-FFT	$10 \times 10$	179.676	6764160	14897664	73728
Fully unrolled 2D-FFT	$10 \times 10$	65.664	1562496	1352484	7680
4-bit TF quantized 2D-FFT	$10 \times 10$	65.028	1711104	1351716	0

The implemented design is a part of the technical specifications for the MEO scenario with 1500 MHz bandwidth and a RF chain size of  $10 \times 10$ . In order to approximate the power and area consumption, the results presented in Table I are extrapolated. The extrapolated area-power estimation for the three FFT implementations is presented in Table II. Considering xcvu29p-l2fsga2577e ultra scale+ FPGA, it is impractical to prototype on a single FPGA using 2D-FFT conventional FFT due to LUT and DSP constraints. The fully unrolled 2D-FFT consumes too many DSP blocks which are required for other processing blocks in a real-time beamformer. The proposed 4-bit TF quantized 2D-FFT has less power consumption and consumes zero DSP blocks, and we can say that the proposed 4-bit TF quantized 2D-FFT is the feasibility of fully digital beamforming in satellite communication systems.

## V. CONCLUSION

This work developed an efficient digital beamforming technique for satellite communications. Firstly, the computational complexity of the conventional FFT was discussed, showing that it processes one input sample per cycle of operation in a rolled fashion. Then, a fully unrolled FFT that processes  $N$  input samples per clock cycle of operation was selected for beamforming. The implemented fully unrolled FFT suffers from high area and power consumption. We therefore proposed and implemented an area-power efficient 4-bit TF quantized 2D-FFT. The implemented truncation in 2D-FFT assisted in area reduction at a loss of 1-bit precision. The implementation methodology with the optimization techniques and pipelining helped in both reducing the area and increasing the maximum operating frequency of the design. The 3D directivity pattern with the proposed 4-bit TF quantized 2D-FFT was also discussed. Further, the FPGA implementation validation results with timing simulations were provided and compared. The resulting lower power consumption and area utilization indicate that the proposed solution is promising for satellite communications.

## REFERENCES

- [1] O. Kodheli et al., "Satellite communications in the new space era: A survey and future challenges," *IEEE Commun. Surv. Tut.*, vol. 23, no. 1, pp. 70-109, Firstquart. 2021.
- [2] A. Arora, C. G. Tsinos, B. Shankar Mysore R, S. Chatzinotas, and B. Ottersten, "Analog beamforming with antenna selection for large-scale antenna arrays," in *Proc. IEEE Int. Conf. Acoust., Speech and Sig. Process. (ICASSP)*, Toronto, ON, Canada, pp. 4795-4799, Jun. 2021.
- [3] X. Zhai, X. Chen, J. Xu, and D. W. Kwan Ng, "Hybrid beamforming for massive MIMO over-the-air computation," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2737-2751, Apr. 2021.
- [4] I. Ahmed, et al, "A survey on hybrid beamforming techniques in 5G: Architecture and system model perspectives," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3060-3097, Fourthquart. 2018.
- [5] P. Angeletti and M. Lisi, "Digital beam-forming network with reduced complexity and low power consumption for array antennas," in *Proc. 21<sup>st</sup> Ka and Broadband Commun. Conf.*, 2015.
- [6] D. Suarez, R. J Cintra, F. M Bayer, A. Sengupta, S. Kulasekera, and A. Madanayake, "Multi-beam RF aperture using multiplierless FFT approximation," *Electronics Lett.*, vol. 50, no. 24, pp. 1788-1790, Nov. 2014.
- [7] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," in *IEEE Spectrum*, vol. 4, no. 12, pp. 63-70, Dec. 1967.

- [8] Fast Fourier Transform v9.1, (2022), LogiCORE IP Product Guide Vivado Design Suite PG109.
- [9] S. H. Mirfarshbafan, S. Taner, and C. Studer, "SMUL-FFT: A streaming multiplierless fast Fourier transform," *IEEE Trans. Circuits. Syst. II: Express Briefs*, vol. 68, no. 5, pp. 1715-1719, May 2021.
- [10] R. Palisetty et al., "Area-power analysis of FFT based digital beamforming for GEO, MEO, and LEO scenarios," in *Proc. IEEE Veh. Technol. Conf. (VTC) Spring*, Helsinki, Finland, pp. 1-5, Jun. 2022.