Ultra Low Latency Hardware Optimised Radix-4 FFT for Optical Wireless FPGA Transceivers via Hermitian Symmetry Characteristics

Michael Codd*, Ciara McDonald*, Yiyue Jiang†, Chunan Chen†, Holger Claussen‡§††, Miriam Leeser† and John Dooley*
*Department of Electronic Engineering, Maynooth University
†Department of Electronic Engineering, Northeastern University (USA)
‡Wireless Communications, Tyndall National Institute
§School of Computer Science and Information Technology, University College Cork
††School of Engineering, Trinity College Dublin
Email:
*michael.codd.2018@mumail.ie, *ciara.mcdonald.2013@mumail.ie, †jiang.yiy@northeastern.edu, †chen.chuna@northeastern.edu,
‡§††holger.claussen@tyndall.ie, †mel@coe.neu.edu, *john.dooley@mu.ie

***Abstract*—Future telecommunication networks are expected to deliver exponential performance increases across all domains and with the increased prevalence of real-time IoT devices, greater emphasis is placed on reducing the latency of network links. Traditionally, wireless networking requirements have been fulfilled primarily through use of the RF spectrum, which is rapidly approaching saturation and will eventually very likely become insufficient to meet all future network demands. The optical spectrum however offers enormous amounts of unrestricted and unallocated bandwidth. Efficient high-modulation indoor LED lighting fixtures could potentially integrate with and complement the RF spectrum for short to medium distance low latency applications. The Fast Fourier Transform (FFT) is a ubiquitous operation in many communication network topologies. Typically the FFT is computed via serial methods which are optimised for low resource usage, however these architectures fall short of the Ultra Low Latency (ULL) requirements for optical wireless communication. Fully parallel FFT computations can achieve nanosecond latency and tens of gigasamples of throughput, far surpassing serial methods. However, their prohibitively high resource utilisation has limited their practical use. In this work, we introduce a hardware optimised, fully parallel architecture for optical wireless communication which leverages hermitian symmetry characteristics within real-valued optical signals and properties of the discrete DFT to reduce the footprint of a fully parallel FFT on an FPGA. The final architecture is implemented on an AMD RFSoC2x2 and requires only 3 clock cycles to compute a 256-point real-valued FFT, a 290 fold reduction compared to an equivalent serial model. The design was tested at 122.88 MHz, resulting in a 24 nanosecond latency, demonstrating its potential for use in optical wireless communication and other high-performance 5G+ networks.**

***Index Terms*—Fast Fourier Transform, Hardware Optimisation, FPGA, Low Latency Communication, Optical Wireless Communication**

The research behind this publication was conducted with the financial support of Science Foundation Ireland (SFI) frontiers program under Grant Number 20/FFP-P/8901

I. Introduction

THE need for Ultra Low Latency (ULL) architectures has grown substantially over the past two decades, driven primarily by the explosive increase in the number of real-time Internet of Things (IoT) devices. Remote access/control or time-critical, multi-endpoint synchronisation applications require reliable data transfer and could potentially benefit much more from a sub-1ms ULL link over 100 Gbit/s of data throughput. The RF spectrum has also become very congested, with limited available bandwidth driving up operating costs. Achieving ultra-high data rates (10s to 100s of Gbit/s) at higher RF bands presents significant challenges for RF engineers, such as propagation path loss or increased signal processing complexity. The optical spectrum encompasses a vast range of frequencies, from infrared (IR) to ultraviolet (UV), and also includes the visible light spectrum. Recent research demonstrated that arrays of solid-state lighting fixtures are capable of gigabit-per-second data transmission using Light Emitting Diodes (LEDs) as the source and photodetectors as receivers [1]. Thus, optical wireless systems can be integrated inexpensively with existing LED lighting elements in modern buildings, requiring minimal extra infrastructure to implement. Data transmission using LEDs is limited by one key drawback: they are an incoherent source of light, which bears no discernible phase information. This limits modulation techniques to being both positive and real-valued. A number of single-carrier techniques exist for optical communication, such as on-off keying, pulse amplitude modulation, and pulse width modulation. However, the dispersive nature of the optical channel requires more complex equalisation techniques with longer convergence times, increasing latency. Real-valued multi-carrier Orthogonal Frequency Domain Multiplexing (OFDM) based techniques, such as asymmetrically clipped optical ACO-OFDM [2], DC offset DCO-OFDM [3], and unitary checkerboard preceded UCP-OFDM [4], are tailored models of OFDM commonly used in RF to take advantage of its inherent multi-path resistance and less complex equalisation that only requires single-tap equalisers. The aforementioned OFDM models infer Hermitian symmetry of the data frames, producing an FFT that is symmetric, with each side equal to the complex conjugate of the other [5]. This can be leveraged to reduce the computational complexity and the number of operations within the digital signal processing components of optical wireless transceivers. In [6], this behaviour is used to reduce the overall number of sub-carriers within the signal to increase spectral efficiency. The primary focus of this work is on the Fast Fourier Transform (FFT) and its inverse (IFFT).

TABLE I
CLOCK CYCLE LATENCY OF RADIX-2 AND RADIX-4 MODELS OF XILINX FFT V9.1 IP VS. HYPOTHETICAL PARALLEL EQUIVALENTS

FFT size	R-2 Serial	R-4 Serial	R-2 Parallel	R-4 Parallel
16	141	82	4	2
64	429	248	6	3
256	1677	859	8	4
1024	7341	3438	10	5
4096	32973	14465	12	6

Typically, off-the-shelf FFT solutions, such as the Native FFT v9.1 IP block from AMD [7], infer a serial computation of the FFT with latency in the hundreds to thousands of clock cycles. Parallel computation of the FFT, where all samples are read and processed simultaneously, significantly reduces the clock cycle latency (see Table I). While serial computation of the FFT achieves the lowest resource usage, full parallel architectures are capable of tens of gigasamples of throughput with sub-100-nanosecond latency that equivalent serial models are incapable of producing with current FPGA fabric speeds. A fully parallel architecture requires just one extra clock cycle per additional radix stage but consumes exponentially more resources as the FFT size increases due to the growing number of stages and number of butterflies/dragonflies in each stage [8]. This is a prohibitive factor for implementations of full parallel architectures in FPGA applications [9], [10]. In [11], a 256-point 5-bit input full parallel FFT is demonstrated but required over 200,000 Look-Up Tables (LUTs). Our research details a full parallel FFT for optical wireless communication with a number of optimisation strategies that significantly reduce fabric resources. In summary, the contributions of this paper are:

- A 3 clock cycle latency, 256-point 31Gsps FFT for IM/DD systems capable of running on relatively inexpensive hardware, such as the AMD Xilinx RFSoC2x2.
- The minimisation of FPGA resource usage through leveraging Hermitian symmetry within the real-valued DFT.
- Experimentally validated solutions compared with floating point references.
- A proposed architecture which is validated at a global clock frequency of 122.88MHz to achieve a latency of 24 nanoseconds.

II. BACKGROUND

A. Ultra Low Latency by Parallel Computation

FPGA fabric allows for concurrent parallel operations that massively reduce latency and increase data throughput. Many applications have taken advantage of this, including [12] and [13], which demonstrate parallel FPGA implementations of pulse amplitude modulation transmitters and receivers. These works achieve massive amounts of data throughput that an equivalent serial model would be incapable of, with the fastest operational clock frequency of FPGAs being only in the 100s of MHz [12]. Parallel FFT implementations have demonstrated very low latency and high throughput architectures, albeit with a prohibitively high resource cost [8], [11].

By utilising the simplified serial processes of mutually independent functions, F, such as the operations within a FFT, and running them concurrently, the latency of a system can be greatly reduced by the number of paths, P, available. A sub-400ns optical wireless transmitter instantiated on an AMD RFSoC2x2 has been demonstrated [14], where the number of clock cycles, CC, per process is sliced from a fully serial system to a fraction of the clock cycles which is determined by P in a partially parallelised design. For any parallelised design with P paths, this equates to:

$$CC_{total} = \left\lceil \frac{\sum_{n=1}^{N} CC_F}{P} \right\rceil, \tag{1}$$

where CC_F is the number of clock cycles for any given n^{th} sample of N functions, F. The needs of a design can be scaled to the hardware requirements of the specified platform, but there is a trade-off between latency, resource utilisation and precision. [15] explores this relationship for FPGA implementations of FFTs while presenting a flexible radix algorithm that achieves a 14% decrease in latency compared to the Xilinx FFT IP core albeit with an increased resource cost. These kinds of trade-offs are ubiquitous within the field of FPGA application engineering whereby any increase in performance or accuracy usually comes with a larger resource cost or when some accuracy is sacrificed to conserve resources. In this paper, this trade-off is investigated while targeting ULL in full parallel FFT architectures without incurring massive resource costs or consequential reduction in accuracy.

B. Radix-4 Fast Fourier Transform

$$\begin{gathered} X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \\ W_N = e^{-j\frac{2\pi}{N}} \end{gathered} \tag{2}$$

The formula for calculating an N-point DFT, defined in (2), requires N^2 complex multiplications and $N(N-1)$ complex additions. This N^2 complexity causes larger DFTs to become very computationally expensive. The Cooley-Tukey FFT [16] is a highly efficient technique for reducing the number of operations by decimating the DFT into multiple shorter DFTs (see Fig. 1 and Fig. 2).

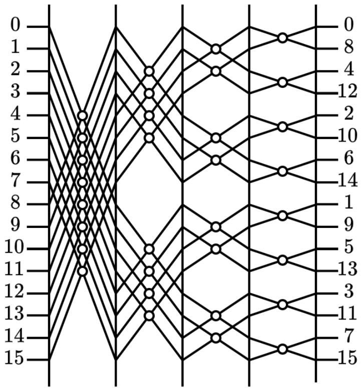

Fig. 1. Radix-2 Vs Radix-4 stages for 16-Point DIF FFT

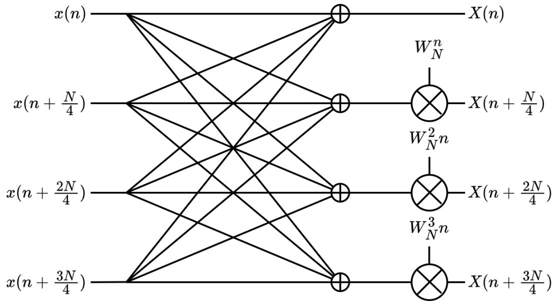

Fig. 2. Radix-4 Dragonfly for DIF FFT

The FFT also leverages simplifications, such as the periodicity of W_N, often referred to as the twiddle factor matrix, to reduce complexity. The decimation of the DFT is typically in orders of 2, 4, 8, etc. Higher order radix DFTs decrease computation cycles but increase complexity and resource usage.

Decimation is the process of breaking down the DFT into smaller DFT's whereby A Radix-N FFT is broken down into N point DFT's or butterfly/dragonflies (see Fig. 2 for radix-4 dragonfly). This decimation is conducted using two main approaches, decimation in time (DIT) or decimation in frequency (DIF). For a DIT FFT, the input time-domain indices are in bit reversed order, while in a DIF FFT, the output frequency-domain indices are in bit reversed order (see Fig. 1). The order in which the butterfly/dragonfly stages are connected is also reversed. Both the fixed point DIT and DIF suffer from quantisation noise deriving from finite word length effects, but due to the different order of calculation, the DIT and DIF FFT produce differing noise effects. The DIT algorithm produces an almost symmetrical error spectrum while a DIF FFT results in an error spectrum that is asymmetrical with increased errors in its lower half [17]. The individual errors of the DIT method are larger than that of a DIF calculation with the asymmetry weighing the error towards the lower half producing a more accurate upper half (see Fig. 3). In certain applications, such as real-valued signals with symmetric DFTs, it may be advantageous to use a conjugate mirror of the upper half of a DIF FFT discarding lower half. A DIF approach is chosen here due to its versatility and higher overall accuracy.

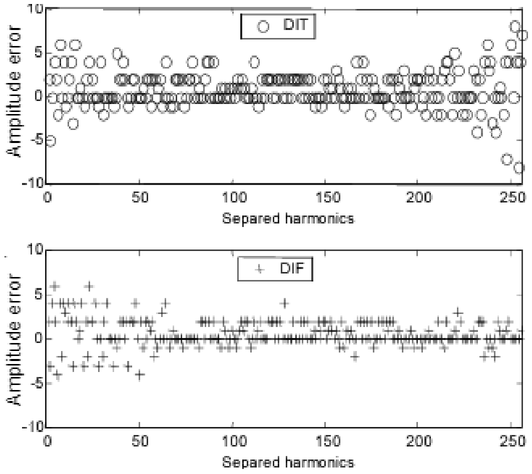

Fig. 3. Errors Harmonics for DIT and DIF FFT, Szolik et al. [17]

III. Hardware Design

The 256-point radix-4 FFT this paper details is implemented on the AMD RFSoC2x2 FPGA. Traditionally, serial implementations of radix FFTs instantiate and reuse one single butterfly/dragonfly for all FFT stages [7]. For fully parallel FFTs, every dragonfly in every stage is instantiated simultaneously. The number of dragonflies D_{count} in an N-point fully parallel radix-4 FFT is

$$D_{count} = \frac{log_4 N \times N}{4} \tag{3}$$

The very large number of dragonflies necessitates careful selection of any internal parameters or input-output word lengths to ensure optimum accuracy while not over utilising the fabric with redundant or unnecessary register width. This section details the hardware architecture of a 256-point full parallel radix-4 FFT including A) The bit resolution of the twiddle factor expansion, B) Internal register width selection and C) A process of back scaling at the output of each dragonfly stage, curtailing the register word width expansion throughout the FFT. A simple low latency rounding compensation of the truncation induced by back scaling is also demonstrated.

A. Twiddle Factors

The twiddle factor matrix, or the expansion/phase factors for an N-point FFT defined in Eq. (4) are the multiplicative coefficients within the FFT. The sequential, compound fixed-point multiplication increases the required internal register widths to retain the output after each stage. Longer expansion factor word lengths provide increased precision but require increasingly more fabric resources, especially with full parallel architectures wherein all dragonflies are instantiated simultaneously. An expansion factor word length of 10 bits equivalent to the FFT input word length was chosen and provided a mean absolute error of $1.4e^{-3}$ compared to a floating point reference twiddle factor matrix.

$$W_N^{kn} = e^{\frac{-2\pi i}{N}kn} \tag{4}$$

B. Internal Register Widths

Similar to the twiddle factors, to ensure most efficient use of resources, any internal register widths should be selected appropriately such that no redundant bit width is allocated. For an N-point DFT with a fixed point input sample word length n bits, ω_{max} the maximum possible transform value is:

$$\omega_{max} = N(2^n - 1) \tag{5}$$

A real-valued signal has a symmetric DFT mirrored about DC. Thus in this case the maximum transform value will appear as two mirrored impulses with:

$$\omega_{max} = \frac{N(2^n - 1)}{2} \tag{6}$$

Translating this to the required register width, From Eq. (6) and, for simplification, allowing $2^n - 1$, the maximum value of any n length fixed point word equal to 2^n, a generalised equation to calculate the output width O_{width} required for an N-point real-valued DFT with input sample word length I_{width} such that O_{width} should not overflow, is defined as:

$$O_{width} = log_2(N) + I_{width} - 1 \tag{7}$$

As shown in Fig. 2, the dragonfly is essentially a miniature 4-point DFT. Using Eq. (7) the 4-point dragonfly DFT requires only one extra bit from input to output. The dragonflies in each stage of an FFT however are scaled by the twiddle factors. The expansion the twiddle factor multiplication has on the required output register width of the dragonflies within a multiplication stage is equal to the addition of the twiddle factor word length. From Eq. (7), for a 4-point dragonfly DFT with twiddle factor word length, W_{width}, the output register word length, O_{width}, for a multiplication stage dragonfly is:

$$O_{width} = 1 + I_{width} + W_{width} \tag{8}$$

The expansion within the parallel FFT induces considerably more resource usage with each sequential stage requiring all internal registers in every dragonfly to grow by $(W_{width} + 1)$. In this work W_{width} = 10 equating to 11 bits per stage.

C. Post Multiplication Stage Back Scaling

Scaling and truncation is an operation used to conserve resources in many FPGA applications. A full scale expansion of the parallel FFT achieves the most accurate fixed point transform. However, the design consumes massive amounts of fabric resources. To reduce the expansion of the twiddle factor multiplication, back scaling is applied by dividing all output samples of each multiplication stage by the maximum absolute value in the twiddle factor matrix. Dividing by this value normalises the scale of the signal equivalent to a floating point FFT. A smaller back scale allowing for larger expansion can provide greater precision but requires more resources. The maximum value however remains the maximum absolute twiddle factor as anything larger over scales the signal greatly reducing precision.

Back scaling by division does not translate well to hardware however, especially when targeting ULL and low resource requirements. A right bit shift of S for division by 2^S is an appropriate substitution. Using Eq. (8), O_{width} for a dragonfly with I_{width}, W_{width}, and now S, the bit shift equivalent to dividing by the maximum absolute value of a signed twiddle factor matrix with n bit word length for $S = n - 1$ is

$$O_{width} = I_{width} + W_{width} - S + 1. \tag{9}$$

In this paper, with $W_{width} = 10$ and $S = 9$, the expansion of the input width I_{width} to O_{width} of the dragonflies stages has been reduced to 2 bits. The final stage of a radix-4 FFT only consists of addition so no back scale here is necessary. Bit shifting is still not a perfect equivalent to division however as it truncates the remainder bits, imitating a divide then floor operation which induced abnormal behaviour around DC and considerably exaggerates the lower half errors of the DIF FFT (See Fig. 3 and Fig. 8). When scaling back at each stage, any small rounding error can propagate via chained truncation and multiplications, magnifying throughout the FFT and thus reducing overall accuracy. Rounding methods such as convergent rounding is traditionally employed at a cost of some latency. A very simple method of rounding can be achieved by leveraging the inherent format of 2's complement notation, by adding the single most significant bit of the truncated remainder after bit shifting. The only case for this method where rounding occurs in the wrong direction is where division results in exactly a negative one-half remainder. While this method of fixed point rounding has a very small bias towards rounding up, the effect is largely negligible and has the added benefit of not requiring any extra clock cycles to compute and only induces a small resource overhead cost.

IV. Experimental Results

A number of architecture models were implemented on the AMD ZYNQ UltraScale+ RFSoC2x2 XCZU28DR FPGA with a global clock frequency of 122.88MHz. The performance of each 256-point FFT model is validated in terms of Normalised Mean Square Error (NMSE) between the output of the on-chip fixed-point FFT and a floating point reference in MATLAB for a set of 20, 256-point 10 bit word length input samples.

A. Architecture Model Details

This design computes each stage in one clock cycle with the final addition/subtraction stage consolidated with the last multiplication stage for a 3 clock cycle latency of the 256-point FFT. This latter option could potentially impact timing closure for larger designs however. In addition, with the instantiation of all dragonflies in each stage, the architecture also allows for continuous throughput with each stage operating concurrently with a new FFT read and output produced every clock cycle providing a throughput of 31GSps at 122.88MHz (see Eq. (10)) [18].

$$\text{CC} = \frac{(Sample_{IN})(Sample_{RATE})}{T_{CLK}} \tag{10}$$

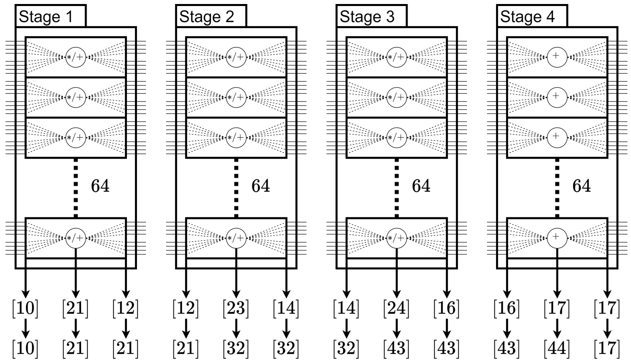

Fig. 4. Dragonfly stage schematic with internal register widths, Top row = scaled, Bottom row = full scale with scale back at last stage

The hardware test bench shown in Fig. 5 is composed of three main modules; A generator block that cycles through 20 256-point data frames. The FFT module itself is instantiated with configurable parameters for scale back intensity and rounding type (more detail on these below). Finally, a combiner at the FFT output is also included that concatenates the output FFT samples into single real and imaginary bit streams for probing convenience. Other modules include a control module which sends appropriate start and reset signals to each of the sub-modules and the Xilinx Integrated Logic Analyser for probing the input and output samples.

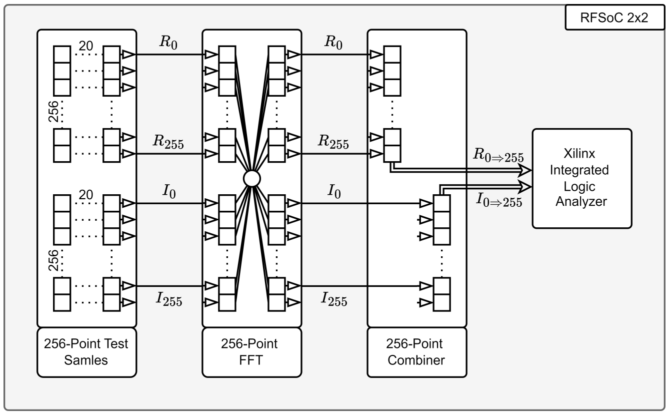

Fig. 5. Hardware Test bench for RFSoc 2x2

B. Real-time Hardware Results

In this paper, 4 FFT models are investigated:

- FS: Full Scale expansion with back scaling at the very last stage of the FFT.
- SB-NC: Scale Back at every stage with no truncation compensation.
- SB-WC: Scale Back at every stage with truncation compensation.
- SB-MNC: Scale Back at every stage with no truncation compensation except the conjugate of the more accurate upper half is now mirrored over lower half at output of the FFT.

All FFT models have a 256-point 10 bit input and 17 bit output resolution derived from Eq. (7). Each FFT stage contains 64 dragonflies with 4 stages. The per stage register width growth defined by Eq. (9) for full scale and scale back models is 11 and 2 bits, respectively (also shown in Fig 4). A summary of the resource utilisation and average NMSE for the set of 20 frames for each model is detailed in TABLE II and Fig. 6. Block Ram utilisation not illustrated in figure for parallel models is zero as twiddle factors are instead stored in registers. The LUT count for the radix-4 serial method* is obtained from a synthesized Xilinx v9.1 FFT model with similar parameter configurations for input and twiddle factor word length that would achieve equivalent fixed point NMSE accuracy. All parallel computation models achieve a 290 fold reduction in latency compared to the serial method. The LUT usage of the FS model is significantly more than the serial implementation, but the scaled models (SB-NC, SB-WC and SB-MNC) reduce this to less than half while still retaining the same latency and comparable NMSE error.

TABLE II
RESOURCE UTILISATION, CLOCK CYCLE LATENCY AND AVERAGE NMSE OF DIFFERENT FFT MODELS

Method	CC	LUTs	NMSE
R-4 Serial*	871	1200	-
Full Scale	3	145841	1.9×10^-6
Scale NC	3	59625	8.7×10^-6
Scale WC	3	67006	3.7×10^-6
Scale MNC	3	59962	3.9×10^-6

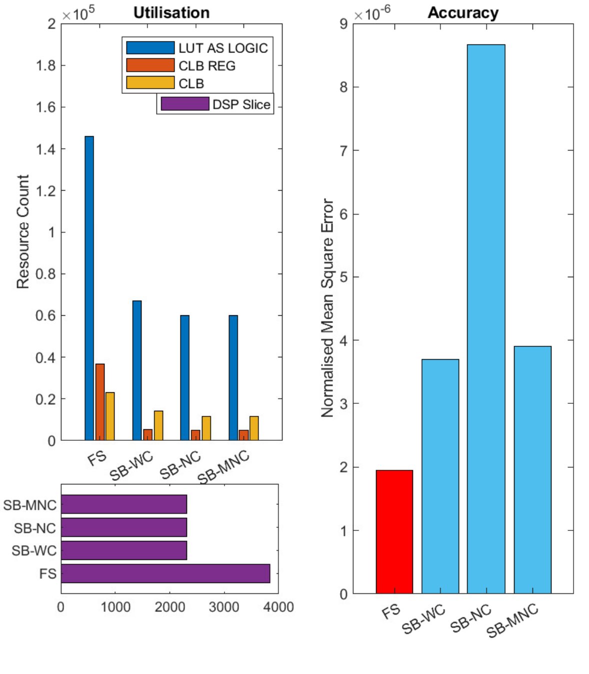

Fig. 6. FPGA resource utilisation of parallel models Vs. accuracy compared to floating point references

One important point of note is that while the FS model completed implementation, routing congestion caused the design to fail to meet timing requirements resulting in a non-functional real-time hardware model. The LUT count is taken from the implementation report and the NMSE of the FS model is calculated using the HDL simulated results and thus serves only as a reference. All other scale-back models passed timing conditions and were implemented in hardware.

SB-NC has a much higher NMSE compared to SB-WC and SB-MNC due to the truncation exaggerating the lower half errors of the DIF FFT. A comparison of the NMSE of the upper and lower halves separately for each model more clearly illustrates this phenomenon (See Fig 7). SB-WC alleviates the increased lower half errors equalising the signal. Fig. 8 compares the sample error between the compensated and uncompensated back scaled models. Fig 7 also illustrates that the upper half of SB-NC NMSE is almost equivalent to the full width SB-WC NMSE. For a real-valued signal with symmetric DFT such as the case with optical wireless signals, SB-MNC can be used instead of SB-WC which takes the SB-NC model discards the lower half and mirrors the complex conjugate of the more accurate upper half at the output instead of using truncation compensation which achieves almost the same level of accuracy as SB-WC but without the extra resource cost overhead.

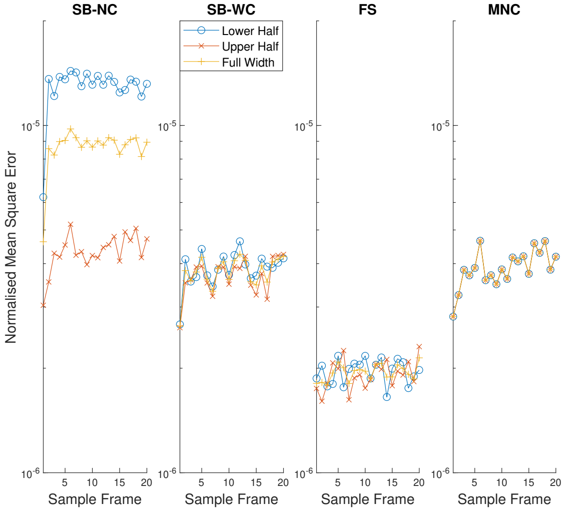

Fig. 7. Upper and Lower Half NMSE Error For SB-NC, SB-WC and FS models

While this model is designed with real-valued optical signals in mind, it can be easily modified for complex input signals by allowing the maximum DFT transform value equal to Eq. (5) instead of Eq. (6), which after successive elaboration via Eq 7-9, infers one extra bit of growth due to the asymmetry within complex input FFTs. SB-MNC would not usable in this case as symmetry is no longer inherent limiting options to SB-NC, SB-WC or FS models.

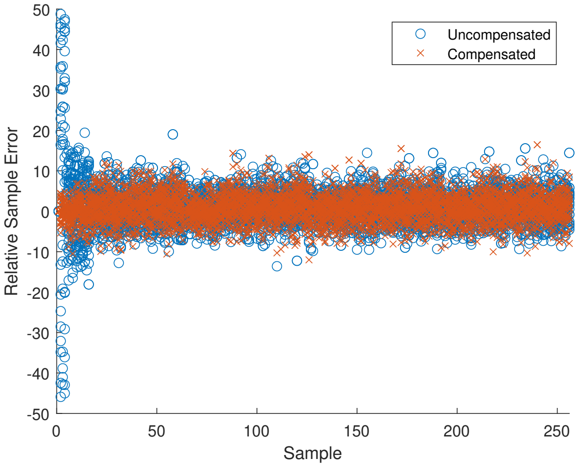

Fig. 8. Back Scale Induced Exacerbation of Lower Half Asymmetry Error Vs. Back Scale with Truncation Compensation

V. Conclusion

The increasing demand for ULL networks necessitates more parallel signal processing methods arising due to current speeds of FPGA fabric limiting serial processing models from achieving the required latency and data throughput speeds necessary for future-proofing such networks. Fully parallel architectures offer extremely high speeds, but they have traditionally come with significant resource costs, making them less feasible for widespread use. This paper demonstrates a 256-point FFT that leverages hermitian symmetry within multi carrier optical wireless signals and properties of the discrete DFT along with a number of simple optimization techniques to reduce resource costs by more than half for a full parallel FFT while maintaining requisite precision. The work presented three potential FFT models: SB-NC scale back without truncation compensation, SB-WC scale back with truncation compensation, and SB-MNC scale back with mirrored upper half instead of truncation compensation for real-valued optical signals. These models can achieve a data rate of 31 gigasamples per second with a latency of just three clock cycles. Importantly, these high-performance optical FFT models are implementable on relatively inexpensive FPGA hardware, such as the AMD RFSoC2x2. By showcasing these models, the paper highlights the potential for highly efficient, low-latency, and cost-effective parallel signal processing solutions for optical wireless communication, paving the way for more advanced and accessible ULL network technologies in the future.

Acknowledgments

This work was supported in part by MathWorks. The authors would like to thank members of the Reconfigurable Computing Lab at Northeastern University as well as colleagues at Maynooth University for useful discussions.

References

[1] Z. Wei, Z. Wang, J. Zhang, Q. Li, J. Zhang, and H. Fu, "Evolution of optical wireless communication for b5g/6g," *Progress in Quantum Electronics*, vol. 83, p. 100398, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0079672722000246

[2] J. Armstrong and A. J. Lowery, "Power efficient optical ofdm," *Electronics Letters*, vol. 42, pp. 370–372, 2006. [Online]. Available: https://api.semanticscholar.org/CorpusID:44142725

[3] J. Carruthers and J. Kahn, "Multiple-subcarrier modulation for nondirected wireless infrared communication," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 3, pp. 538–546, 1996.

[4] T. E. Abrudan, S. Kucera, and H. Claussen, "Unitary checkerboard precoded ofdm for low-papr optical wireless communications," *Journal of Optical Communications and Networking*, vol. 14, no. 4, pp. 153–164, 2022.

[5] Zhang, Fuzhen, *Hermitian Matrices*. New York, NY: Springer New York, 2011, pp. 253–292. [Online]. Available: https://doi.org/10.1007/978-1-4614-1099-7$_8$

[6] C. McDonald, H. Claussen, R. Farrell, and J. Dooley, "Improved spectral efficiency for optical wireless communications using hermitian symmetry characteristics," in *19th RIA/URSI Research Colloquium on Radio Science and Communications*, 2022.

[7] "Fast fourier transform v9.1 logicore ip product guide," Amd Xilinx, 2022. [Online]. Available: https://docs.amd.com/r/en-US/pg109-xfft/Fast-Fourier-Transform-v9.1-LogiCORE-IP-Product-Guide

[8] I. A. Hernández and J. A. López, "Any-radix efficient fully-parallel implementation of the fast fourier transform on fpgas," in *2023 38th Conference on Design of Circuits and Integrated Systems (DCIS)*, 2023, pp. 1–6.

[9] X. Zou, Y. Liu, Y. Zhang, P. Liu, F. Li, and Y. Wu, "Fpga implementation of full parallel and pipelined fft," in *2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, 2012, pp. 1–4.

[10] S. Zhou, X. Wang, J. Ji, and Y. Wang, "Design and implementation of a 1024-point high-speed fft processor based on the fpga," in *2013 6th International Congress on Image and Signal Processing (CISP)*, vol. 2, 2013, pp. 1112–1116.

[11] G. Polat, S. Ozturk, and M. Yakut, "Design and implementation of 256-point radix-4 100 gbit/s fft algorithm into fpga for high-speed applications," *ETRI Journal*, vol. 37, no. 4, pp. 667–676, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.4218/etrij.15.0114.0678

[12] L. Chen, C. Li, C. W. Oh, and A. T. Koonen, "A low-latency real-time pam-4 receiver enabled by deep-parallel technique," *Optics Communications*, vol. 508, p. 127836, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030401821009858

[13] Y. Wang, Y. Chen, L. Zhang, S. Xu, K. Wang, and J. Yu, "Fpga implementation for 44.2368-gbit/s pam8 signal transmission with pruned pre-equalization," *Opt. Lett.*, vol. 48, no. 17, pp. 4562–4565, Sep 2023. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-48-17-4562

[14] C. McDonald, T. E. Abrudan, F. Cabral, S. Kucera, H. Claussen, R. Farrell, and J. Dooley, "FPGA Implementation of a sub-400ns 6G Free-Space Optical Wireless Communications Transmitter," *Opt. Express*, vol. 31, no. 16, pp. 25 933–25 942, Jul 2023. [Online]. Available: https://opg.optica.org/oe/abstract.cfm?URI=oe-31-16-25933

[15] G. Inggs, D. Thomas, and S. Winberg, "Exploring the latency-resource trade-off for the discrete fourier transform on the fpga," in *22nd International Conference on Field Programmable Logic and Applications (FPL)*, 2012, pp. 695–698.

[16] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, 1965, uRL: `http://cr.yp.to/bib/entries.html#1965/cooley`.

[17] I. Szolik, K. Kovac, and V. Smiesko, "Influence of digital signal processing on precision of power quality parameters measurement," *Measurement Science Review*, vol. 3, no. 1, pp. 35–38, 2003.

[18] S. Agarwal, S. R. Ahamed, A. Gogoi, and G. Trivedi, "A 28-gbps radix-16, 512-point fft processor-based continuous streaming ofdm for wigig," *Circuits Syst. Signal Process.*, vol. 41, no. 5, p. 2871–2897, may 2022. [Online]. Available: https://doi.org/10.1007/s00034-021-01917-0