Low-Latency FFT/iFFT RTL Implementation for the FALCON Post-Quantum Signature Algorithm

Alexandre Ortega, Lilian Bossuet and Brice Colombier
Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
{alexandre.ortega; lilian.bossuet; b.colombier}@univ-st-etienne.fr

I. Introduction

FALCON [1] is one of the three post-quantum digital signature schemes that have been recently standardized by NIST due to the future threat that quantum computers pose to classical cryptographic schemes. Despite this, there is currently no full hardware register-transfer level (RTL) implementation of FALCON. One possible explanation is the rather unusual requirement for a double-precision floating-point Fast Fourier Transform (FFT) [2], which is used in FALCON to speed up polynomial multiplication. In this work, we propose a full RTL constant-time implementation of the FFT and its inverse (iFFT), on FPGA, tailored for the specific context of FALCON. Section II presents the FFT in the context of FALCON before Section III describes the proposed architecture. Section IV concludes.

II. The Fast Fourier Transform in FALCON

In FALCON, the FFT over the ring $\mathbb{Q}[x]/(\phi)$ is used with $\phi = x^N + 1$ and $N = 2^k$ a power of two. N is a security parameter of FALCON that can be equal to either 512 or 1024. Due to FALCON security requirements, IEEE-754 compliant double-precision floating-point arithmetic is being used [1]. Using the fact that FALCON polynomials are in $\mathbb{Z}[x]/(\phi)$, as well as the roots of unity symmetry in $\mathbb{Z}[x]/(\phi)$, the storage requirements can be halved and more than half of the computations can be omitted [1].

III. Description of the proposed Hardware Architecture

The proposed hardware architecture is divided in three main blocks.

1) The butterfly unit: Made with three complex double-precision floating-point operators, an adder as well as a subtractor and a multiplier, it can be reconfigured dynamically to perform either the radix-2 decimation-in-time FFT or the radix-2 decimation-in-frequency iFFT.

2) The Polynomial coefficients storage unit: Two true dual port RAMs are used to store the polynomial coefficients.

3) The Polynomial Coefficient Addresses and Root of Unity Values Storage Unit: Four single-port ROMs and one dual-port ROM are used to store the pre-computed coefficient addresses in RAM and root of unity values.

The proposed hardware implementation is described using VHDL and synthetised using AMD-Xilinx Vivado 2023.2.

TABLE I
(i)FFT-512/(i)FFT-1024 Implementation Results

	This work		[3]	Vivado 2023.2	
Floating-point precision	**Double**		Double	Single	
FFT length	**512**	**1024**	512	512	1024
LUT	**9658**	**9677**	8396	1741	1793
FF	**369**	**374**	2526	3468	3508
DSP	**36**	**36**	9	10	10
BRAM	**8**	**11**	9.5	4	5
Latency (cycles)	**3074**	**6658**	19800	4589	9474

Table I reports the implementation results of the proposed design and compares it with Vivado 2023.2 FFT/iFFT IP, and a co-design implementation of FALCON FFT/iFFT for the security parameter $N = 512$ [3]. The FPGA targetted, in all the reported results in Table I, is the AMD-Xilinx ZCU104+ (xczu7ev-ffvc1156-2-e) FPGA. The fairest comparison is with Mandal et al. design as both design use double-precision. The proposed design and this design have similar metrics for the LUTs and the BRAMs. The proposed design uses $4\times$ more DSP blocks but around $6.5\times$ less FFs and clock cycles. As expected when comparing the proposed design to Vivado's IP, which uses single-precision, it uses around $2\times$ more BRAMs, more LUTs and DSPs. However, the proposed design uses around $10\times$ less FFs and achieves a lower latency.

IV. Conclusion

A low-latency full hardware constant-time RTL implementation of the FFT/iFFT, tailored for FALCON parameters, was presented. It achieves the best latency of the literature among FPGA-based implementations.

Acknowledgment: This work received funding from the France 2030 program, managed by the French National Research Agency under grant agreement No. ANR-22-PETQ-0008 PQ-TLS

References

[1] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang, *et al.*, "Falcon: Fast-Fourier lattice-based compact signatures over NTRU," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, pp. 1–75, 2018.

[2] L. Beckwith, D. T. Nguyen, and K. Gaj, "High-Performance Hardware Implementation of Lattice-Based Digital Signatures." Cryptology ePrint Archive, Paper 2022/217, 2022.

[3] S. Mandal and D. Roy, "Design of a Lightweight Fast Fourier Transformation for FALCON using Hardware-Software Co-Design," in *GLSVLSI'24 Proceedings*, pp. 228–232, 06 2024.