Precision Unwound: Fine-Tuning Loop Unrolling for Energy-efficient FPGA-based PQC using HLS

***Abstract*—High-Level Synthesis (HLS) is a valuable tool for designing hardware accelerators for Post-quantum Cryptography (PQCs). However, while HLS can efficiently map high-level code to hardware, the quality of the synthesized hardware in terms of latency, power, and area is sensitive to various design parameters and configurations, such as loop unrolling, pipelining, and dataflow optimizations. In this work, we explore the effects of loop unrolling on the execution time and the energy efficiency of the final PQC. We demonstrate that, despite initial expectations, loop unrolling could worsen the performance.**

I. Introduction

PQC standards were announced by NIST in September 2024 and are available as C/C++ code. HLS is a valuable tool for designing FPGA-based PQC hardware accelerators using C/C++ as a design entry. The quality of the synthesized hardware in terms of latency, power, and area remains highly sensitive to various design parameters and configurations. In HLS, loop unrolling is a common optimization technique where iterations of a loop are duplicated to allow for parallel execution, improving performance by reducing latency. This approach can significantly reduce the latency of loop-intensive applications by allowing each iteration of the loop to operate independently on dedicated hardware resources.

II. Unrolling Factor versus Energy Efficiency

A classical approach to determine the fastest hardware configuration is to unroll the loop to the maximum extent. This is valid in many situations, where for example the loop is regular without any memory dependency. This technique is particularly effective in cases where the loop is regular and free from memory dependencies, meaning that each loop iteration does not depend on the results of previous iterations and can access memory independently without conflicts. However, an increase in the initiation interval (II) of the loop can introduce performance degradation, nullifying the effect of the increased parallelism. The relation between the loop unrolling factor and the initiation interval determines the performance and energy efficiency of the final hardware.

Our experiments have been performed on the Number Theoretic Transform (NTT) module that executes the polynomial multiplication in lattice-based PQC standards. We used Altera HLS and Quartus Prime v24.01 to compile the optimized NTT code (on List 1) proposed by [1] generating designs with different unrolling ratios on FPGA Cyclone-10GX 10CX220YF780I5G at 100MHz, to avoid other timing issues affecting our analysis (results on Fig. 1).

Listing 1. *NTT*, refactored code

```
k = 1;
zeta = zetas[k++];
for (len = 128; len >= 2; len >>= 1){
   limit = len; start = 0;
   for (int j = 0; j < 128; j++){
   #pragma unroll
      uint16_t r_j_len = r[start + len];
      uint16_t r_j = r[start];
      t = fqmul(zeta, r_j_len);
      r[start + len] = r_j - t;
      r[start] = r_j + t;
      start++;
      if (start == limit){
         start += len;
         limit += (len << 1);
         zeta = zetas[k++]; }}}
```

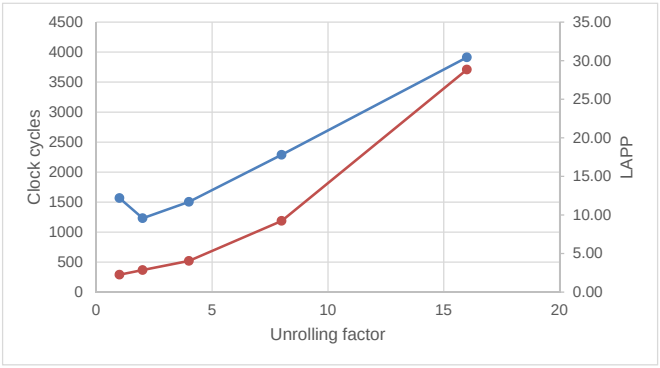

Fig. 1. Unrolling factor versus clock cycles and LAPP (Latency-Area-Power-Product). The results show that a minimum clock cycle is achieved with an unrolling factor of 2 (blue line), after which performance decreases linearly. Similarly, the effects on LAPP are also similar, yielding the best results even without unrolling.

III. Conclusion and Future Work

In this work, we present a preliminary study on the effects of loop unrolling on performance and the energy efficiency of PQC. We demonstrated how synthesis directives such as loop unrolling might lead to sub-optimal results. Future work includes the study of the analytical methods to predict these effects before synthesis, as well as the trade-offs introduced by different memory configurations. The framework and the full dataset will be released as open-source upon publication.

References

[1] A. Guerrieri, G. D. S. Marques, F. Regazzoni, and A. Upegui, "Optimizing post-quantum cryptography codes for high-level synthesis," in *2022 Euromicro Conference on digital systems Design (DSD22)*, Gran Canaria, Spain, 2022, pp. 361–67.