Evaluation of Optimized PQC Standards ML-KEM and ML-DSA on Sargantana RV64GBV core

Xavier Carril[1,2], Emanuele Parisi[1], Narcís Rodas[1], Raúl Gilabert[1], Juan Antonio Rodriguez[1], Oriol Farràs[3] and Miquel Moretó[1,2]

[1]*Barcelona Supercomputing Center (BSC), Barcelona*
[2]*Universitat Politècnica de Catalunya (UPC), Barcelona*
[3]*Universitat Rovira i Virgili (URV), Tarragona*

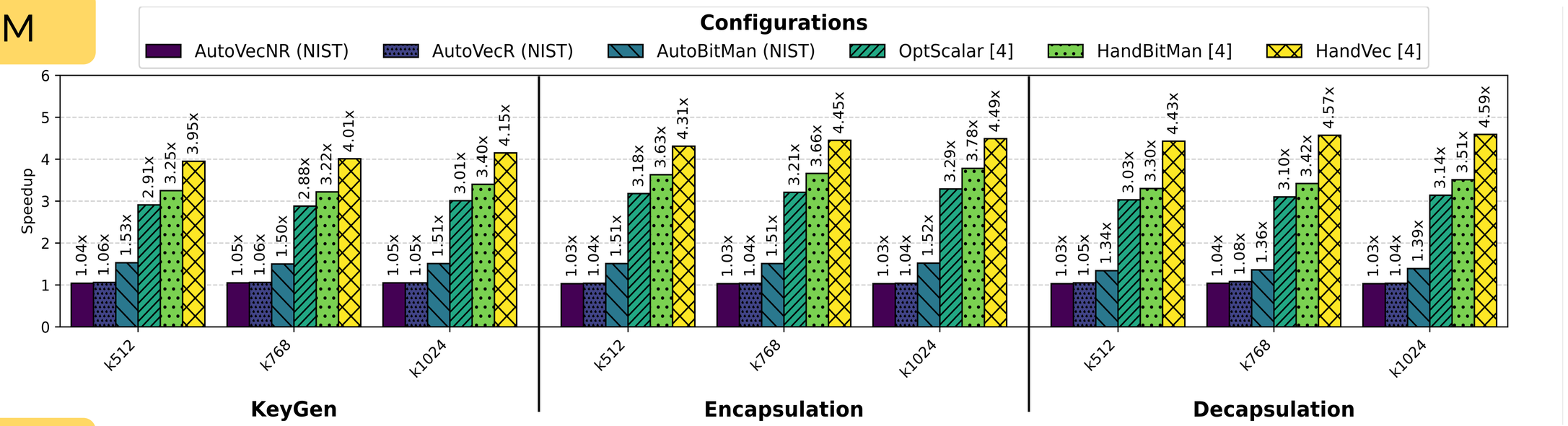
Motivation

- Previous research focus on accelerate Post-Quantum Cryptography (PQC) schemes, using custom ISA extensions [1]
- This work evaluates RISC-V Bit Manipulation (B) and Vector (V) ISA extensions for NIST PQC standards, ML-KEM[2] and ML-DSA[3].
- Comparison between reference and optimized implementations from Zhang et al. [4] using BV extensions.
- Analyze performance gaps between hand-optimized and compiler-generated code (auto-vectorization and auto-bit manipulation).

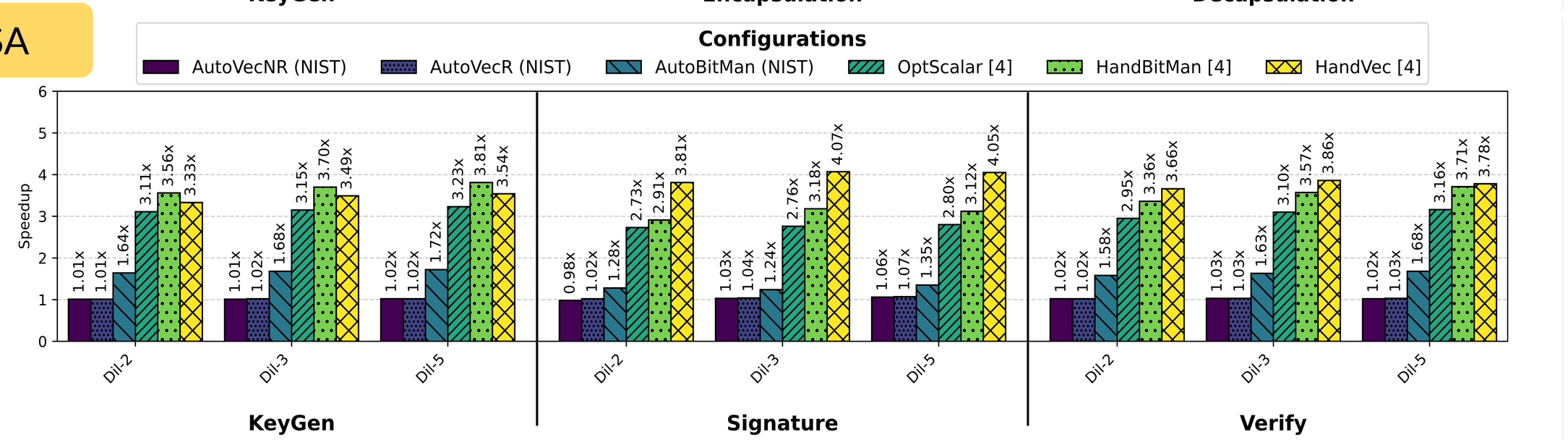
Methodology

- **Hardware:**
  - Single-issue Sargantana RV64GBV core [5]
    - 128-wide SIMD unit supporting RVV1.0, LMUL ≤ 1
    - Support for Bit Manipulation (B extension)
- **Optimized Cryptographic Primitives:**
  - Keccak (SHA3 primitives): Benefits from Bit Manipulation
  - Number Theoretic Transform (NTT): Benefits from Vectorization
- **Compilation and Execution Tools:**
  - Use of gcc 14.2 for hand-optimized and compiler-generated code
  - Use of Xilinx Alveo U55c FPGA at 25MHz clock frequency

ML-KEM



ML-DSA



Configurations

- **AutoVecNR**: Auto-vectorization, no register renaming.
- **AutoVecR**: Auto-vectorization with register renaming.
- **AutoBitMan**: Compiler-generated bit manipulation instructions.
- **OptScalar**: Hand-optimized scalar code [4].
- **HandBitMan**: Hand-optimized bit manipulation code [4].
- **HandVec**: Hand-optimized vector code [4].

Conclusion

- Manual vectorization and bit manipulation significantly outperform compiler-generated optimizations.
- Compiler auto-vectorization shows limited impact against scalar reference version.
- Register renaming gives minimal benefit (~0.02×).
- As Keccak implies >50% of execution cycles, auto-bit manipulation achieves higher speedups than auto-vectorization.
- Full potential of RV64GBV extensions is only realized through manual optimization:
  - Up to **84.3%** improvement (Decapsulation k768) over optimized-scalar code.

[1] Tim Fritzmann et al. “RISQ-V: Tightly Coupled RISC-V Accelerators for Post-Quantum Cryptography”. In: IACR TCHES (2020). doi: 10.13154/tches.v2020.i4.239-280.
[2] National Institute of Standards and Technology (2024) Module-Lattice-Based Key-Encapsulation Mechanism Standard, CSRC. Available at: https://csrc.nist.gov/pubs/fips/203/final
[3] National Institute of Standards and Technology (2024) Module-Lattice-Based Digital Signature Standard, CSRC. Available at: https://csrc.nist.gov/pubs/fips/204/final
[4] Jipeng Zhang et al. “Optimized Software Implementation of Keccak, Kyber, and Dilithium on RV{32,64}IM{B}{V}”. In: IACR TCHES (2024). doi: 10.46586/tches.v2025.i1.632-655.
[5] Víctor Soria-Pardos et al. “Sargantana: A 1 GHz+ InOrder RISC-V Processor with SIMD Vector Extensions in 22nm FD-SOI”. In: 2022 25th DSD. 2022. doi: 10.1109/DSD57027.2022.00042.

Xavier Carril is supported by the predoctoral programme AGAUR-FI Joan Oró grant (2024 FI-1 00520), funded by the Generalitat de Catalunya (Department of Research and Universities) and the European Social Fund Plus. Emanuele Parisi is supported by AI4S fellowships from the “Generación D” initiative (Red.es, Ministerio para la Transformación Digital y de la Función Pública, C005/24-ED CV1), funded by EU NextGenerationEU funds through PRTR, partially funded by Generalitat de Catalunya [2021-SGR-00763], and by Spanish MCIU/AEI project PID2023-146511NB-I00 co-funded by EU ERDF. Oriol Farràs is supported by the grant 2021 SGR 00115, by the project HERMES funded by INCIBE and by the EU NextGeneration EU/PRTR, and the project ACITHEC PID2021-124928NB-I00 funded by MCIN/AEI/10.13039/501100011033/FEDER, EU. This work is supported by the Chips Joint Undertaking (JU), European Union (EU) HORIZON-JU-IA, under grant agreement No. 101140087 (SMARTY).