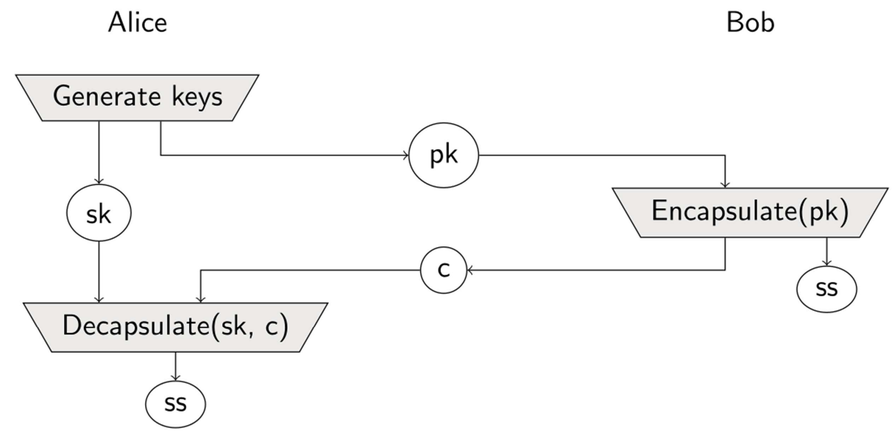
A Unified Yet Efficient Parallel Hardware Architecture for FrodoKEM

Giuseppe Manzoni[1], Aydın Aysu[2] and Elif Bilge Kavun[3]

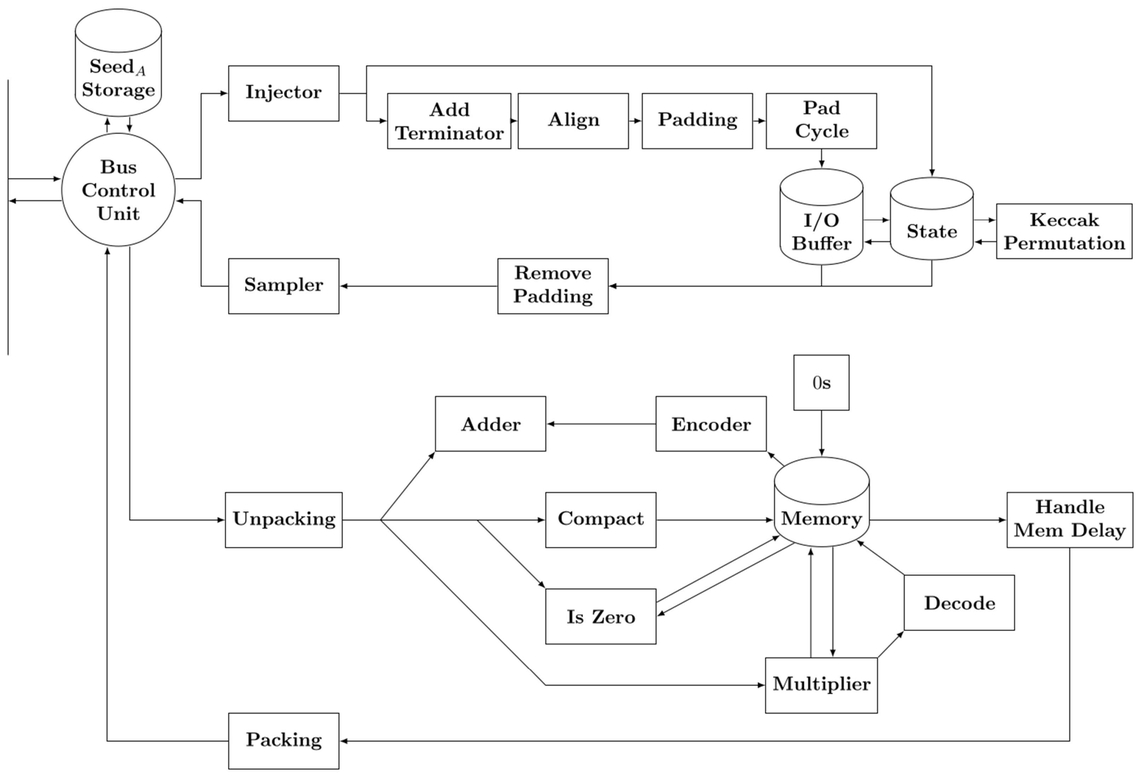
[1] Barkhausen Institut, Dresden, Germany giuseppe.manzoni@barkhauseninstitut.org
[2] North Carolina State University, Raleigh, NC, USA aaysu@ncsu.edu
[3] Barkhausen Institut & TU Dresden, Dresden, Germany elif.kavun@barkhauseninstitut.org

Why FrodoKEM?

- When quantum computers become available, Shor's Algorithm will break conventional public key cryptography we have been using
- FrodoKEM is a Post-Quantum Cryptography (PQC) algorithm to establish a shared secret [1]
  - Based on the unstructured lattice problem, which is generally considered to have a higher security margin than structured lattice
  - Not selected as standard by NIST, but this was not due to security problems [2], yet recommended by Germany [3], France [4], Netherlands [5], Turkey [6], etc.



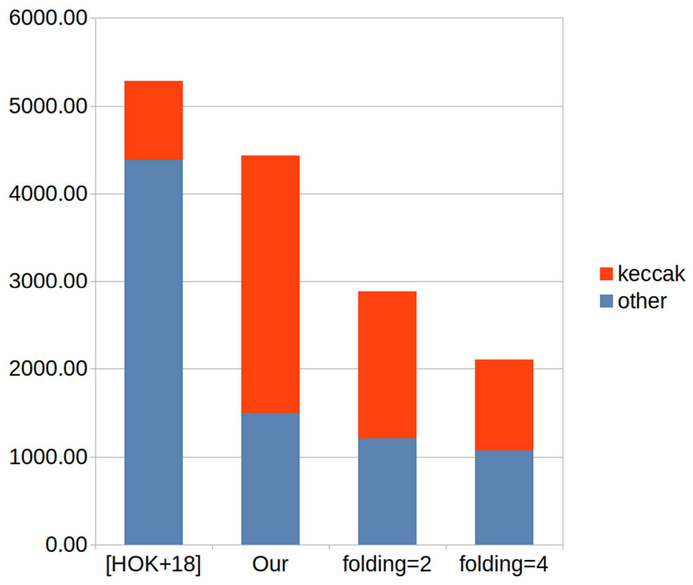
Key Encapsulation Mechanism (KEM)



Our module's overall architecture

Our existing implementation*

- A single module that can carry out **any FrodoKEM algorithm** (key generation, encapsulation, decapsulation) **with any parameter set** (640, 976, 1344), one operation at a time
- Our benchmark is [HOK+18]:
  - They provide 6 different modules each that carries out one specific operation
  - Our module is **overall** 14% smaller and 15x faster (than their largest module)
  - Our module is more flexible, as we can change operation and parameter set at run time



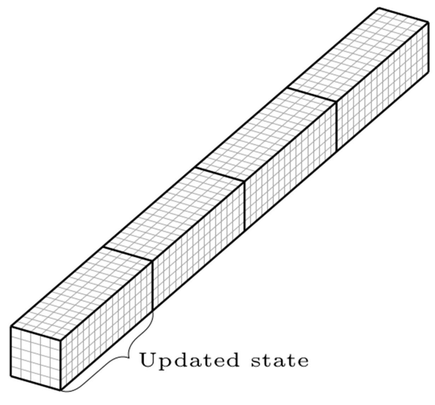
Comparison of existing results with estimates if we add folding to Keccak

Comparison of [HOK+18] and our current work*

- We use less area overall
- Yet our Keccak module is much bigger

<u>This is because our existing module does not use folding, and it executes a permutation per clock cycle</u>

n-folded Keccak implementation allows us to need only 1/n of the update logic, at the cost of being n times slower (needs n clock cycles to complete)



A folded implementation updates only a fragment of the state and shifts it every clock cycle

Comparison of speed and area of:

- Our current work* for different parameter set configurations
- [HOK+18], either for the decapsulation module (.D) or for a set of modules to have comparable functionality
- Our estimates for our planned next steps, using a folded Keccak implementation, for 2 and 4 folds

References

[1] https://frodokem.org/files/FrodoKEM_standard_proposal_20241205.pdf
[2] https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf
[3] https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf?__blob=publicationFile&v=9
[4] https://cyber.gouv.fr/sites/default/files/2022/04/anssi-avis-migration-vers-la-cryptographie-post-quantique.pdf
[5] https://www.ncsc.nl/binaries/ncsc/documenten/publicaties/2022/juli/guidelines-for-quantum-safe-transport-layer-encryption/guidelines-for-quantum-safe-transport-layer-encryption/Guidelines_for_PQC_-_Kyber_archief.pdf
[6] en.bilgem.tubitak.gov.tr/en/on-the-negative-effects-of-quantum-computers-on-secure-communication-and-actions-to-be-taken/

[HOK+18] 'Standard Lattice-Based Key Encapsulation on Embedded Devices' by James Howe, Tobias Oder, Markus Krausz, and Tim Güneysu. Published in TCHES 2018.

* Our existing work has been already presented at LightSEC 2025 and will be soon published as 'An Optimized FrodoKEM Implementation on Reconfigurable Hardware' by Springer. (Authors: Giuseppe Manzoni, Shekoufeh Neisarian and Elif Bilge Kavun)

This work is supported in part by DFG Project No. 543352068 and by NSF Award No. 2350142 and by the AMD University Program.