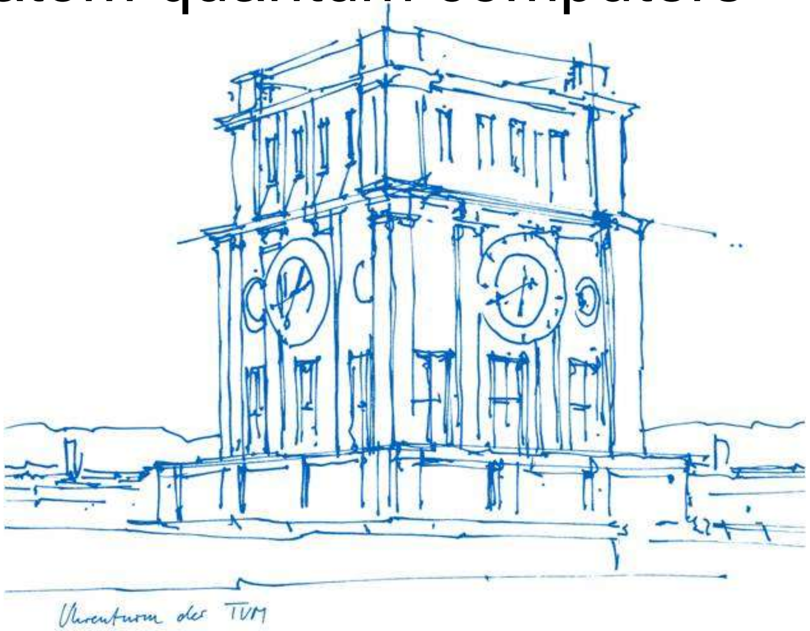
Tooling and benchmarking of a hardware-agnostic compilation toolchain for neutral-atom quantum computers

Emil Khusainov

Adviser: Yanbin Chen

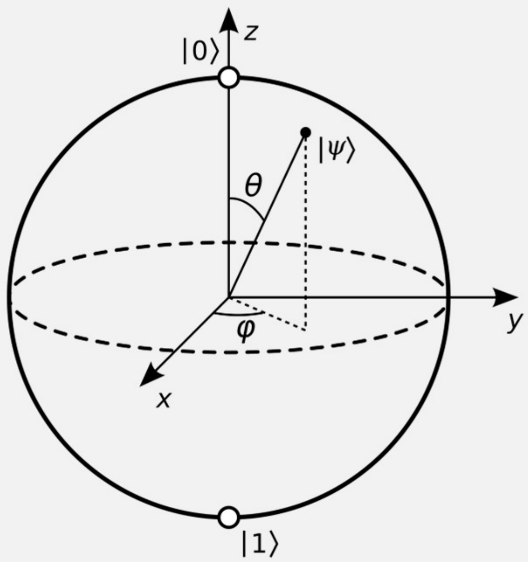
Supervisor: Prof. Dr. Christian B. Mendl

Munich, 04. July 2025



Quantum fundamentals

- Qubit as quantum analogue of classical bit, but with superposition of basis states
- Possible representation as vector on a Bloch-Sphere
- By measurement will collapse in one of the basis state with a certain probability
- Key properties: entanglement, no-cloning, superposition, collapse



Quantum hardware architectures

The most popular

Superconducting Qubits

- Based on nonlinear LC circuits
- QPUs of IBM, Google
- Pro:
  - Fast gates
  - Easy electronic
  - Good scalable and designable
- Con:
  - Short coherence time
  - Necessity in low temperatures

Trapped-Ion Qubits

- Based on individual atoms in electomagnetics traps
- Used by IonQ
- Pro:
  - Very long coherence times
  - High fidelity of gates
- Con:
  - Slow gates
  - Complex system

Quantum hardware architectures

Currently Emerging

Photonic Qubits

- Based on physical states of photons
- Pro:
  - No need in absolute zero temperatures
  - Low loss transmission in quantum networks
- Con:
  - Probabilistic two-qubit gates
  - Requires high-quality hardware
  - Hard scaling

Topological Qubits

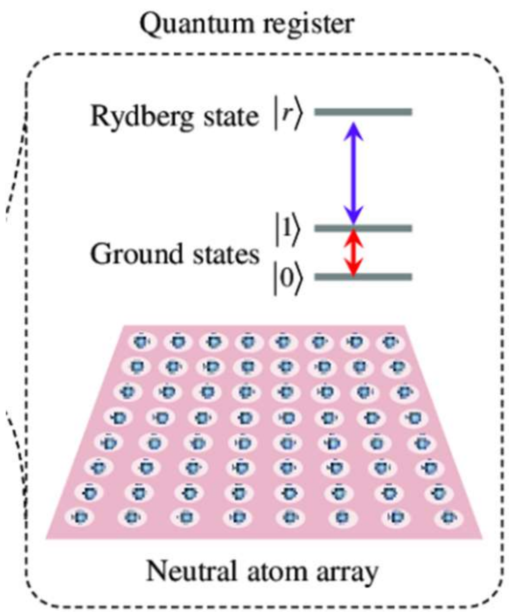
- Based on non-abelian anions with braiding operations
- Hyped Majorana from Microsoft
- Pro:
  - Intrinsic protection against certain errors
  - Potentially large scalable
- Con:
  - Experimental

Quantum hardware architectures

Neutral Atom

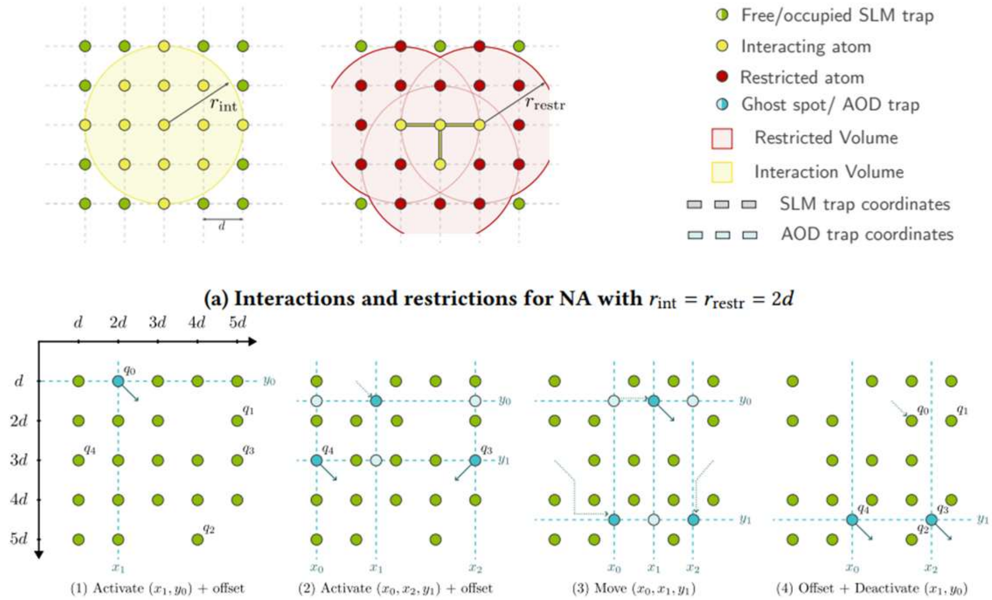
The most interesting for us now

- Based on neutral atoms such as rubidium placed in optical tweezers (SLM)
- Hyperfine states of atoms denotes basis states
- Rydberg interaction allows C_kZ and CZ_k gates
- Rydberg radius => high connectivity
- Long coherence time
- Raman laser for single rotations
- Can be used in DPQA that allows „shuttling“ of qubits in runtime (aka FPDA in classical computing)



Shuttling

Optical tweezer controlled by SLM, but moveable AOD can take atom from 1 SLM and bring it to another



(a) Interactions and restrictions for NA with $r_{\text{int}} = r_{\text{restr}} = 2d$

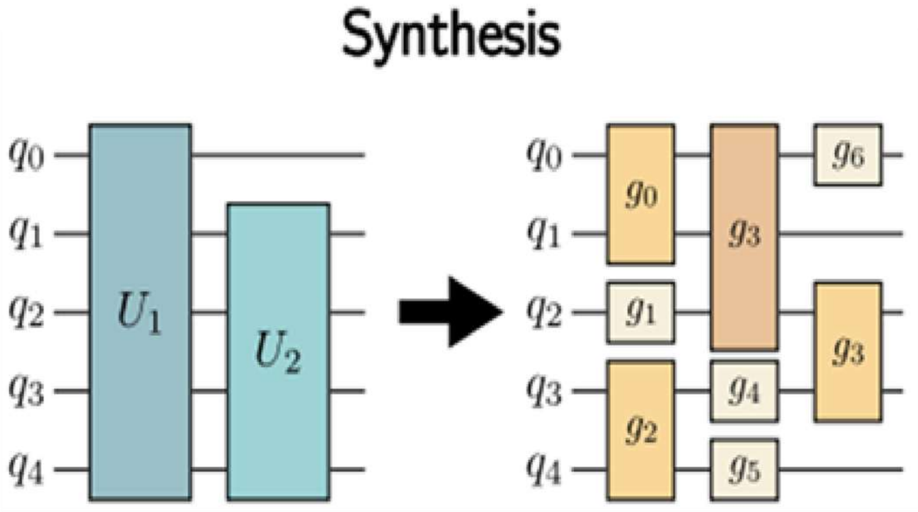
(b) Qubit array shuttling with *Acusto-Optic Deflectors* (AOD) parallelization constraints.

Compilation of quantum circuits

Synthesis

- Goal is to transpile a quantum computation into native gate set of the target hardware
- By Neutral Atoms target gate set can be wide

- Possible Algorithms:
  - KAK-decomposition
  - Euler-decomposition

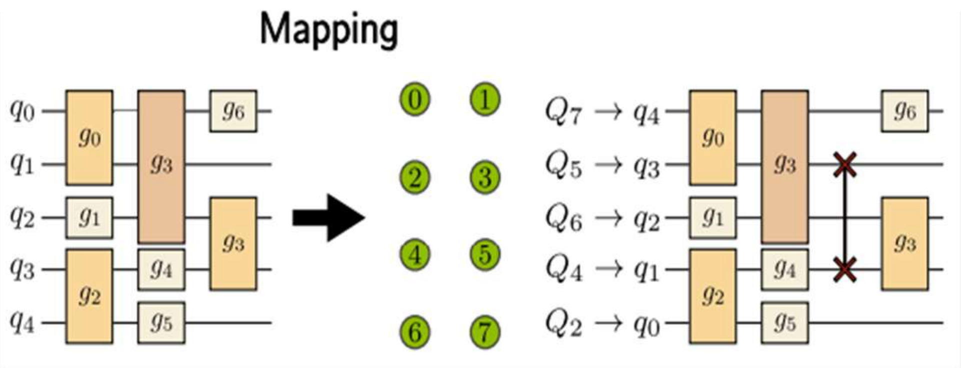


Compilation of quantum circuits

Mapping

- Goal is to insert SWAP gates so that all connection between qubits fullified for next gate
- Here DPQA Shuttling gives another possibilities for mapping but makes it harder
- Interaction radius

- Used Algorithms:
  - SABRE
  - A*

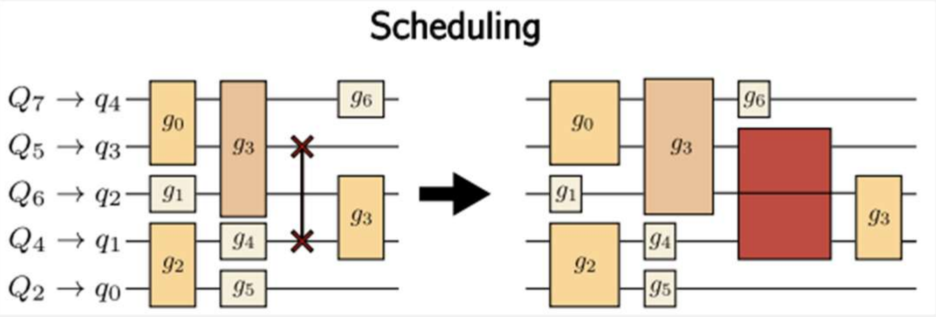


Compilation of quantum circuits

Scheduling

- Goal is to determine which gate should be executed next according to hardware restrictions such as number of lasers, their times, crossing paths

- Possible Alorithms:
  - ASAP
  - ILP



Considered Compilers

HybridMapper from MQT

- Doesn't have synthesis step, but gets all possible gates with Architecture file, Synthesis is done independently preprocessing via Qiskit
- Uses cost function to determine whether to use SWAP or Shuttling
- Has only mapping and scheduling steps, no circuit optimization

Considered Compilers

Enola

- Shuttling only algorithm
- Tries to create a steps where each step has an array where each pair is mutual independent (MIS)
- Doesn't consider number of AODs
- Built-in synthesis using QisKit
- Doesn't consider fidelity of 1 qubits gates
- Has a lot of mismatches -> Later

Considered Compilers

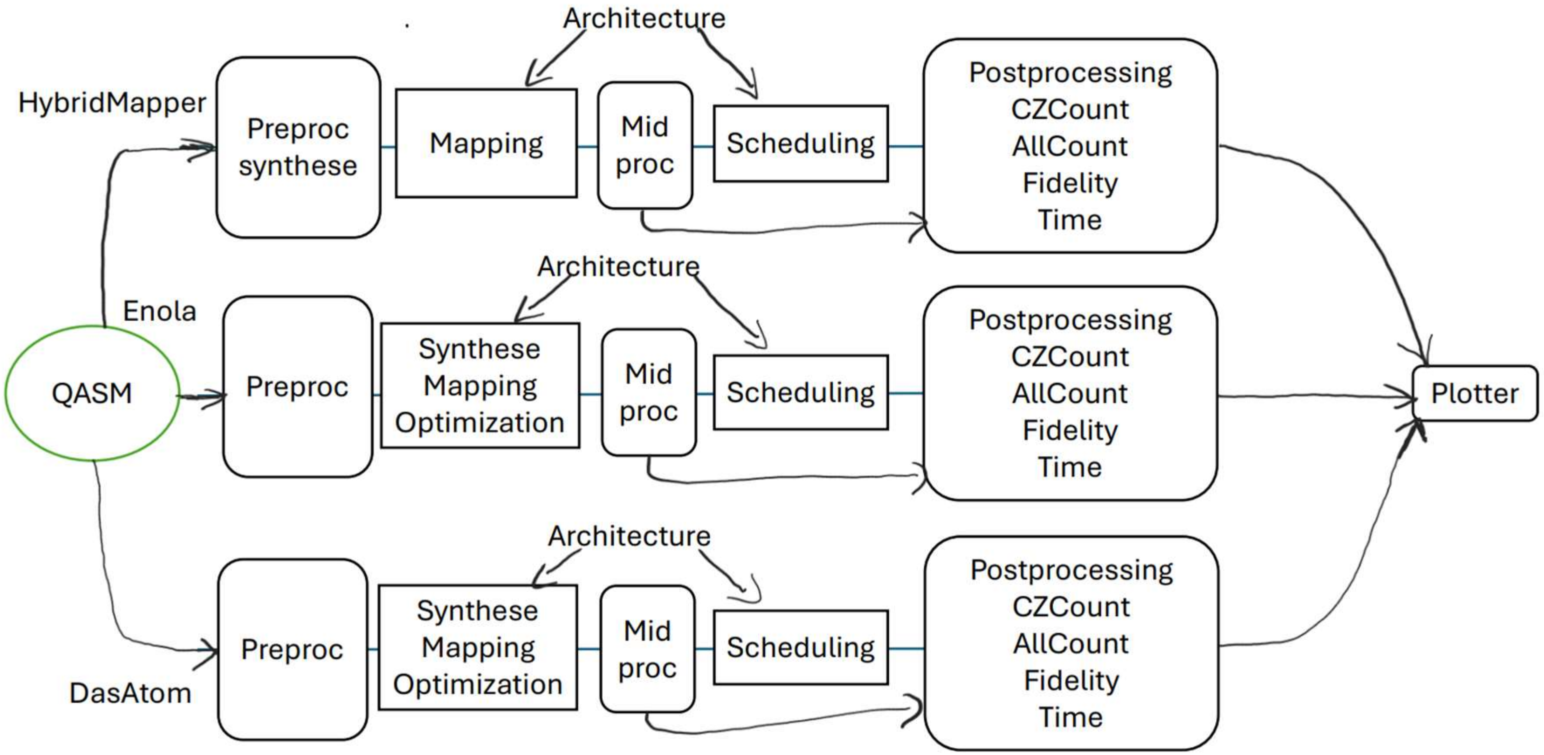
DasAtom

- Based on Enola and Tetris
- Tries to make an independent circuits to minimize Depth (DAC)
- Also completely SWAP-Free
- Same synthesis as by Enola
- Doesn't consider fidelity of 1 qubit gates
- Promises an exponential fidelity outperform over Enola (QFT30 414 Times)

Testing

Testing

Simplified Execution Flow

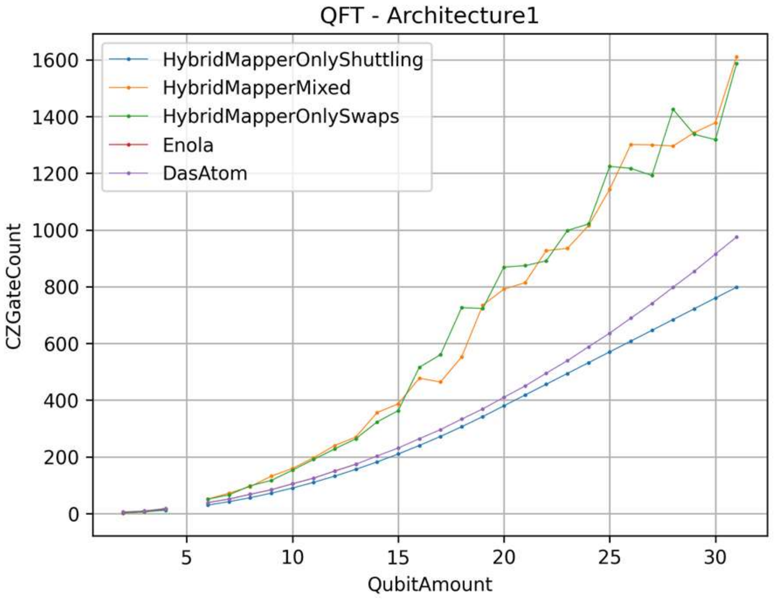


Testing

Interpretation of results

Quantum Fourier Transformation Algorithm 2- 30 Qubits, CZ gate Count

- Swap based mapping adds a lot of CZ gates
- Enola and DasAtom used the same amount of CZ gates and consider it in fidelity
- HybridMapper Shuttling-Only gate count is the best

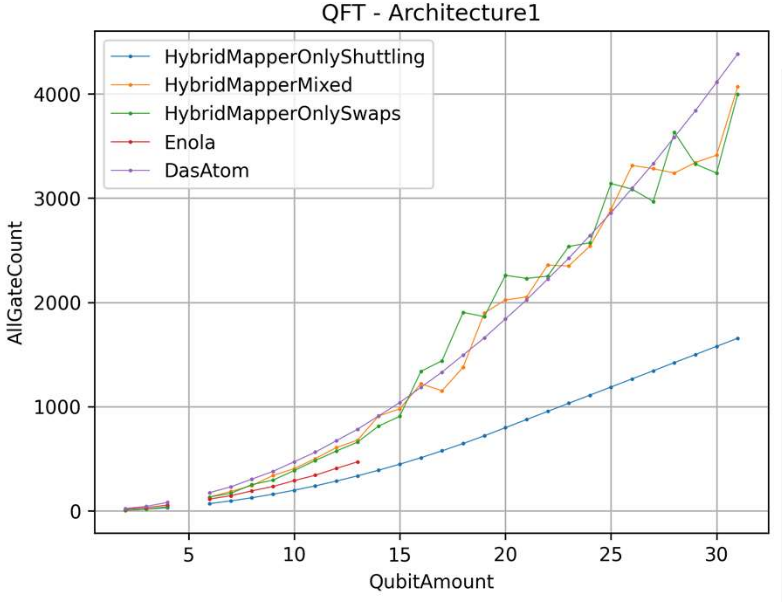


Testing

Interpretation of results

Quantum Fourier Transformation Algorithm 2- 30 Qubits, All gate Count

- HybridMapper Shuttling-Only gate count is still the best
- DasAtom uses much more gates than Enola
- Recap: Enola and DasAtom doesn't consider fidelity of single qubit gates => not fair

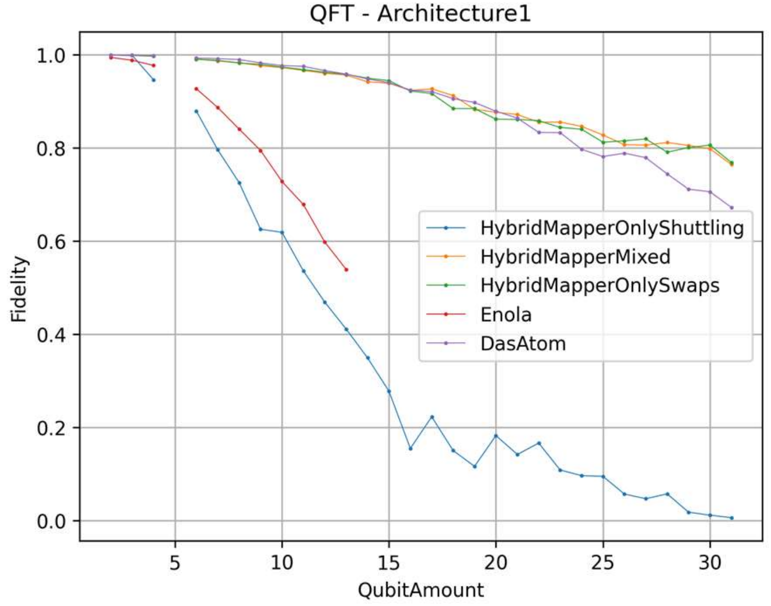


Testing

Interpretation of results

Quantum Fourier Transformation Algorithm 2- 30 Qubits, Fidelity

- Enola and HybridMapper in Shuttling fall fast.
- At QFT30 Enola has fidelity of 0.00089
- Nevertheless, DasAtom has 800 times higher than Enola
- DasAtom states 414x better than Enola
- Not accurate results but indicative



Testing

Example Outputs Enola&DasAtom

Metric/Compiler	Enola	DasAtom
Circuit	qft_indep_qiskit_30.qasm	qft_indep_qiskit_30.qasm
Fidelity Overall	0.0008991	0.7060
Fid. Movement	0.69376	0.9934373
Fid. Coherence	0.00154	0.81603
Gate Count	2370	4111
CZ Gates	915	915
Fid. 1Q	1(-)	1(-)
Compile Time s	14251	2.5

Questions and assumptions

- 1 qubit gate Fidelity impact?
- Why does DasAtom outperform Enola exponentially? → Why is there a huge difference between fidelity components?

- Arising assumption: Check the used metrics system, unify and fix it if something.

Equalization

Equalization

Comparing of calculations

Source of coherence fidelity discrepancies

- DasAtom used correct formula for fidelity of coherence: $fid = e^{\frac{-t_{idle}}{T_{coh}}}$
- Enola used Taylor approximation and possible loss-of-significance of floats: $fid = \prod_{q \in Q}(1 + \frac{-t_{q_{idle}}}{T_{coh}})$
- Now both are using exponential variant
- But the idle time was different

Equalization

Comparing of calculations

Source of time discrepancies

- Different approaches for time of movement calculations
- DasAtom used simplification $t = \frac{dist}{speed}$
- Enola used a Bluvstein et al. model for calculating time $\mathrm{t} = 200\left(\sqrt{\frac{d}{110}}\right)$
- Both approaches differs noticeably when d isn't 110 um
- Now Enola also uses simplificated model
- But distance was also different

Equalization

Comparing of calculations

Source of idle distance discrepancies

- DasAtom distance calculation was fine and made sense
- Enola also…
- However, Enola doesn't consider transmitted Architecture parameters on mapping step
- The problem was an incorrectly implemented transfer of global parameters

```python
def set_hardware_paramters(param: dict):
    R_B = param["R_B"]  # rydberg range
    AOD_SEP = param["AOD_SEP"]  # min AOD separation
    RYD_SEP = param["RYD_SEP"]  # sufficient distance to avoid Rydberg
    SITE_SLMS = param["SITE_SLMS"]  # number of SLMs in a site
    SITE_WIDTH = param["SITE_WIDTH"]  # total width of SLMs in a site
    SLM_SEP = AOD_SEP  # separation of SLMs inside a site
    X_SITE_SEP = RYD_SEP + SITE_WIDTH  # separation of sites in X direction
    Y_SITE_SEP = RYD_SEP  # separation of sites in Y direction
```

Final testing

Final testing

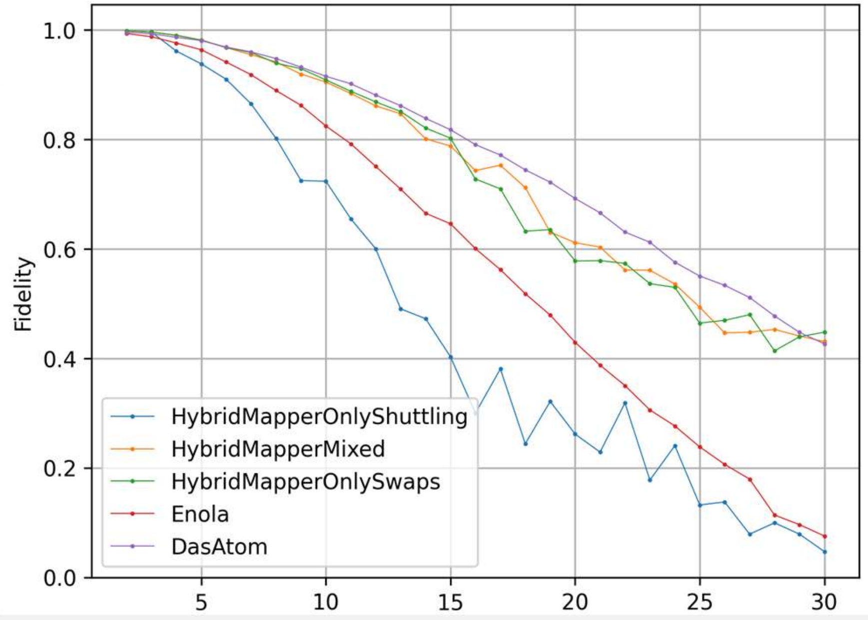
Interpretation of results

Quantum Fourier Transformation Algorithm 2- 30 Qubits, Fidelity

- Added 1QG fid., correct coherence time and distance, architecture parsing and minor simplifications

- Clear differences observed
- E.g. Enola has only 5.5 times lower fidelity than DasAtom

- Nevertheless, there were a few more fidelity sources that Enola considered, but they weren't turned off



Literature

[1] BlockSphere
[2] SLMArray
[3] Hybrid Circuit Mapping
[4] Compiler Development Neutral Atoms
[5] Enola
[6] DasAtom
[7] Bluvstein
[8] GitHub

Thanks for your attention