SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Tooling and Benchmarking of a Hardware-Agnostic Compilation Toolchain For Neutral-Atom Quantum Computers

Emil Khusainov

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Tooling and Benchmarking of a Hardware-Agnostic Compilation Toolchain For Neutral-Atom Quantum Computers

Tooling und Benchmarking einer hardwareunabhängigen Kompilierung Toolchain für Neutral-Atom Quantencomputer

Author:	Emil Khusainov
Examiner:	Prof. Dr. Christian Mendl
Supervisor:	M.Sc. Yanbin Chen
Submission Date:	04.07.2025

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, 04.07.2025 Emil Khusainov

Acknowledgments

I would like to express my sincere gratitude to my Advisor, Yanbin Chen, for his interesting topic and research idea, continuous support and guidance throughout this thesis. Also for regular feedbacks, for directing me to the right path, and for connecting me with other people and universities.

Abstract

Neutral Atom Quantum Computer (NAQC) are new emerging architecture in the world of quantum computations. Its high fidelity, native multi-qubit gates and possibility to use in Dynamically Field-Programmable Qubit Arrays (DPQA) features a multitude of promises. Moreover, DPQA allows using physical shuttling of qubits during circuit execution instead or together with logical SWAP gates. All this opens up new compilation possibilities. Therefore, a new software tools that should take an advantage of additional capabilities of NAQC are needed. The goal of this work is to compare existing tool chains for compilation, that took possibility of qubit shuttling into account. Furthermore, the solutions will be aligned at their points of calculation divergence to enable direct comparison of the algorithms they use. This approach will reveal potential inconsistencies present in early benchmarks as well as in the considered compilers' own original publications.

Contents

1 Introduction

Classical Computing on CPU is not exhaustive [1]. Consequently, scientists introduced coprocessors such as GPUs and FPGAs. They were created to accelerate certain types of tasks, which they successfully did, but now they have reached the limit according to computability theory [1]. Thus, there are a multitude of problems that can not be efficiently solved only by changing and adapting classical CPU architectures. Several of those have not any proven polynomial classical algorithm [1, 13]. Nevertheless, a number of quantum states saved in one quantum register grows exponentially with number of qubits, that allows to solve particular problems in efficient way. Theoretically, this will give such a scale, which is often called quantum advantage [13].

Currently, Quantum Computer (QC) do not outperform classical systems, however they have a considerable potential to do that in the near future [1, 13]. Current state is called Noisy Intermediate Scale Quantum Computing Era (NISQ) which allows exploring a world of quantum entanglements, superpositions, error corrections and quantum information interpretation [15]. To leave this era and go into Fault-Tolerant Quantum Computing Era (FTQC) new ways to correct or not to make errors, to improve coherence time and operating temperatures, together with overall fidelity and speed improvements considered [16].

One of those ways is to explore different architectures for QC, which potentially can solve one or more of above-mentioned constraints. There are already several promising architectures such as Trapped-Ion QC [3], Superconducting QC [6], Topological QC [10] and more. In this work only tools for NAQC will be considered and evaluated [23, 18, 24].

NAQC has several advantages over other architectures [19]. For example a long coherence time of qubits that are based on Rydberg states of Rubidium or Cesium [18]. This allows to make a significantly more calculations before fidelity became unacceptable low. Moreover, NAQC has a rare feature that sets it apart from others. This feature is a possibility to execute gates not in strictly defined positions but everywhere where a Rydberg laser can point [5, 18]. As a result, gates can be applied to several qubits at once implementing a built-in multi-qubit gate support [23, 5, 18, 11]. And the most interesting is that atoms can be physically moved by Acousto-Optic Deflector (AOD) that can be used for separating zones or as another possibility to perform an analogous for SWAP but physically and not logically or moreover consider a combination of those [23, 19].

Taken together, this leads to a considerable number of possibilities for optimizing quantum computations in NISQ and at the same time complexities during compilation [8, 7, 22, 18, 19, 12].

In this work a different tool chains in compilation will be considered, such as HybridMapper (HM) from Munich Quantum Valey (MQT) [19], DasAtom [7] or Enola from UCLA-VAST Lab (UCLA-VAST) [22]. They will be benchmarked, evaluated, then the issues in calculations [4] will be fixed, and the result will be evaluated again. For consistency the used architectures will be approximated as closely as possible between used compilers.

2 Related Work

In this work different compilation toolchains for neutral atom architecture are considered [7, 22, 19]. The global goal is to determine the impact of qubit shuttling onto overall circuit fidelity and amount of gates. Moreover, during this work will be determined drastic mismatches in benchmarks of considered toolchains. And a step in direction to possible fair comparison is done.

As a related work consider [4]. In it author tries to show possible mismatches in results of compilation because of minimal input parameters deviations, often substantial and difficult to rationalize discrepancies. Therefore, raising an important issue of fair evaluation, which is essential for understanding which approaches perform better or worse.

2.1 My Contribution

The idea of this work is to bench different tools for neutral atom compilation to understand what approaches are better and what considered hardware constraints raise significant impact on overall compilation quality. Similarly, it is important to consider the other side of the problem shown by [4]. This aspect involves not only comparing discrepancies resulting by differences in input values, but also to determine the reasons why a minority of these inputs made the compilation process ill-conditioned.

3 Background

3.1 Quantum Basics

3.1.1 Idea of Quantum Computation

For more than 50 years of using classical paradigms of computations it has been recognized, that parallel to classical ones exists also a quantum version of them [2]. Which offers clearly different and perhaps much more powerful features than classical computational theory [13, 1].

3.1.2 Quantum State and its Representation

The following section is primarily based on [14] supplemented by [1, 13].

Quantum mechanics uses operations with states in a complex Hilbert space to perform computations. This quantum state serves as an analog of an ordinary bit in the quantum world. In the following considered only one-qubit system for simplicity w.l.o.g. in so-called computational basis. Moreover, this state possesses not only basis states $|0\rangle$ and $|1\rangle$, but also an infinite amount of combinations of this eigenstates, this is called a quantum superposition. To represent it two complex numbers α and β are used. For one-qubit system we define a quantum state as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

The coefficients must satisfy the following normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1$$

To transfer quantum information into classical bits a measurement operation is performed. Measurement will always collapse a wave function into one of corresponding to result the basis states. The probability of outcome is calculated with Born's rule as follows:

$$P(0) = |\alpha|^2, \quad P(1) = |\beta|^2$$

For visualization, the state can be represented geometrically as vector in Bloch Sphere, where angles θ and ϕ will uniquely determine the position of the vector.

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$$

4 Neutral Atom Architecture

There are already a considerable amount of approaches for building a QC. All of them are based on different physical systems that manage the creation, connection and manipulation of qubits [23]. Several of the promising approaches are:

4.1 Prevalent Architectures

4.1.1 Superconducting Qubits

The main insights into this technology have been collected by [6]. This architecture uses superconducting resonant circuits, which apply microwave signals to control and read qubits.

The strong sites are its very fast single- and two-qubit gates, electronics that cover current needs have been around for a long time and have been well studied e.g. commercial microwave devices can be used in experiments, an increase of qubit amount in moderate effort is possible, different types of qubits 4.1 and parameters with easy coupling nature of qubits allow high designability.

The week sides are necessity in close to absolute zero temperature to operate, strongly limited coherence times, crosstalk between qubits requires careful design, complex error correction mechanisms, a qubit is not true two-level system, thus undesired transitions must be avoided during processing.

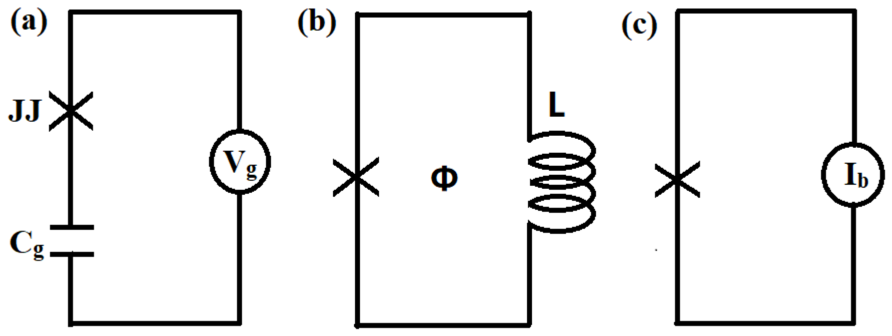


Figure 4.1: Three types of superconducting qubits circuit diagrams [6]

4.1.2 Photons

Key concepts of this technology are drawn from the work [17]. Uses photons as qubits and raise several points; computations are done via beam splitters, phase shifters, and detectors.

To advantages could be assigned operations with interference and entanglement by room temperatures, low decoherence due to weak interaction with environment, could bring a significant potential in the networks of quantum computers in the future.

Meanwhile, disadvantages include necessity in high-quality hardware to save and detect states, complex error correction techniques and hard scaling.

4.1.3 Topological

The main insights into this technology have been collected by [10, 26]. Encodes information non-locally using non-abelian anions such as Majorana zero modes in topological superconductors or exotic materials. Logical gates are implemented via braiding operations which are inherently error-resistant.

Advantages are built-in topological protection which enhances resilience to local noise, potentially reducing error correction complexity.

Disadvantages are hard experimental realization of stable non-abelian anions, several operations are still theoretical or nascent.

4.1.4 Ion Traps

Key concepts of this technology are drawn from the work [3]. Uses positive ions such as Yb or Ca placed in electromagnetic traps. Qubit states are encoded in stable electronic levels.

The strong sites are high coherence time, that can be exceptionally long also without applying of decoupling techniques, high fidelity of one-qubit and two-qubit gates, straightforward state preparation and readout.

The weak side is non-trivial scaling over fifty ions due to heating and crowding processes.

4.2 Neutral Atom

The main considered physical architecture of this paper is NAQC. This type of computer uses laser cooling and trapping techniques for neutral atoms and optical or microwave pulses to manipulate quantum states. It offers long coherence times, scaling in 1D, 2D or even 3D, and fair connectivities by long-range interactions between atoms in Rydberg state [23].

4.2.1 Physical Hardware Implementation

Consider 4.2 to have a schematic overview. Neutral atoms such as Rubidium, Cesium or Strontium are positioned inside an ultra-high vacuum cell in a specific geometric arrangement, typically achieved by optical tweezers. To work with it, it needs to be cooled to low temperatures. The process of cooling is multistaged that especially involves Doppler cooling using a laser. To place cooled atoms on fixed locations and keep them a trap laser controlled by Spatial Light Modulator (SLM) is employed. With help of SLM laser could be focused onto very small configurable points, this effect called optical tweezers. SLM could arrange atoms in 1D, 2D,or even in 3D configuration. To move atoms there are another mobile traps controlled by AOD. AOD implements the comparable to SLM function, but for mobile optical tweezers. Then all electronics should be precisely calibrated to control process run [18, 23].

To perform actually computations laser and microwaves are employed, depends on executed gate used so-called Raman or Rydberg pulses, which could work locally or globally on arrangement of neutral atoms. [18, 23]

4.2.2 Processing

According to [23, 24]. To use neutral atoms for performing quantum computations encoding of qubit computation basis state needs to be determined. These encoding should have a long coherence time and weak interaction with environment. For Rubidium atom it is based on the spin of the furthest electron. Meanwhile, a Strontium implementation a nuclear spin of atom is used.

For implementing a one-qubit rotations operators two laser induced known as Raman transition are used. Target atom exposure to multiple photons and after that emits some another amount of energy, what led to change of it state, by combining the duration, power and frequency of laser different direction and angles of rotation could be achieved.

For implementing a two-qubit gates a so-called Rydberg state is used. In this state an outermost electron has a very huge influence on other atoms' electrons in arrangement. Thus, other atom can not go in Rydberg state. This effect widely known as Rydberg blockade. An effective impact of this effect is an implementation of a Controlled-Z gate.

Moreover, this effect enables the implementation of multiple-controlled Z-Gate directly on hardware, since Rydberg blockade influences not only neighboring qubits but extends as a distance-dependent interaction. The strength of blockade obeys the propor-

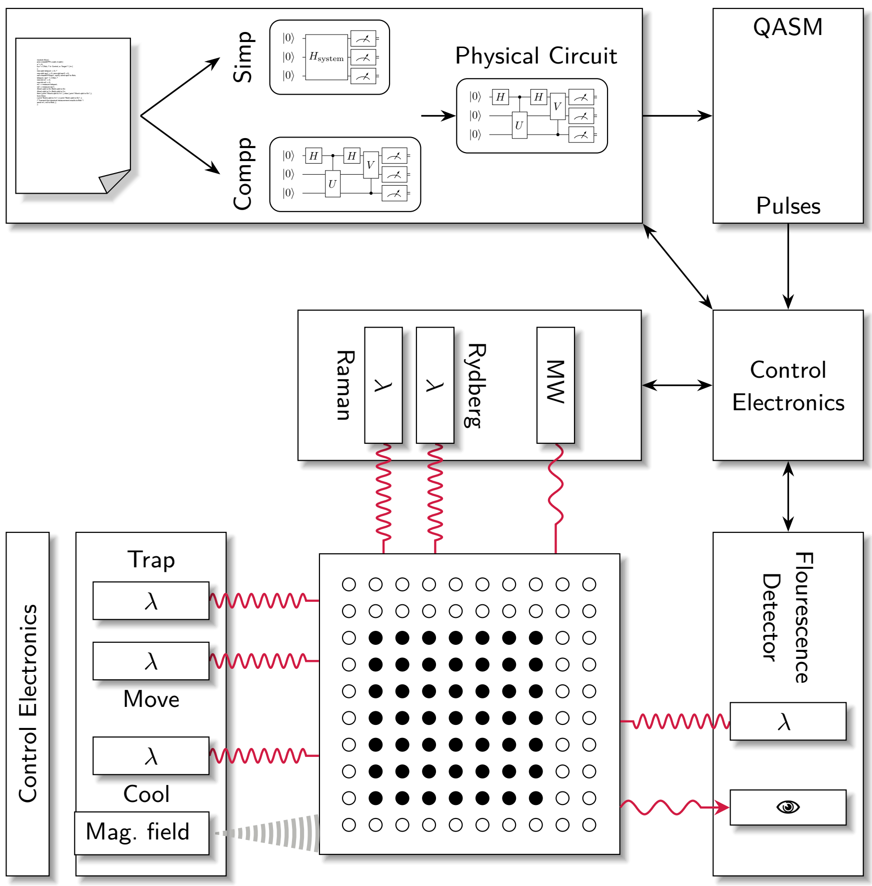


Figure 4.2: Schematic component overview of a cold atom quantum computer [23]

tion for small distances $\frac{1}{|r|^3}$ and $\frac{1}{|r|^6}$ for long. Depending on a coefficient that accepted for blocking and interacting, different multiple-controlled Z-Gate are implemented.

In addition, Rydberg blockade effect is used for long-range interactions between qubits within the acceptable distance, it is with NAQC possible to execute a two-qubit gate and do not require specifically to place the atoms directly next to each other.

Nevertheless, if an interaction radius is small for gate execution, logical SWAP gates are considered to logically move qubit to the target qubit. But DPQA could be based on neutral atoms. Hence, it makes possible to use physical moving of qubit in lattice, by moving it from SLM to AOD and back. This process is known as qubit shuttling.

For visualization of the described features consider 4.3.

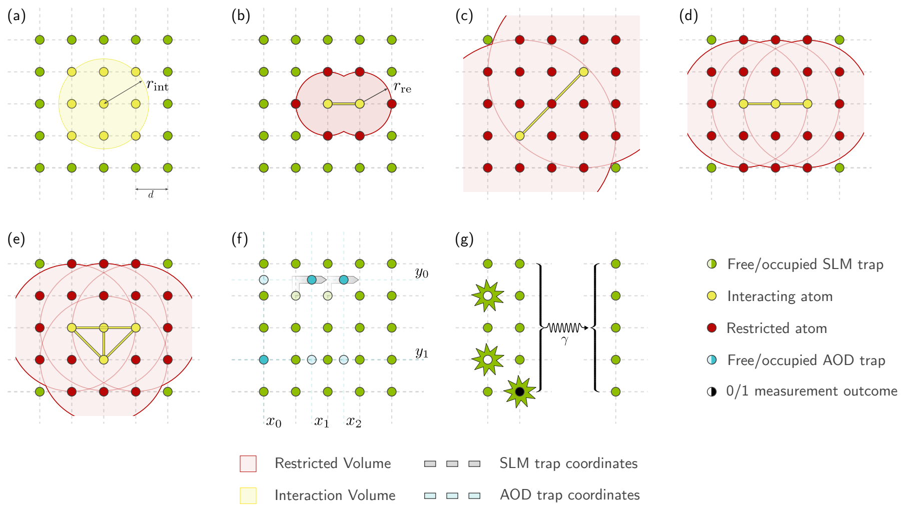


Figure 4.3: **Capabilities of the NAQC platform.** In this setup, atoms are arranged on a regular grid of SLM traps, with a fixed distance denoted as d. **(a)** Rydberg blockade interaction: interacting gates can be performed to all qubits within this range. **(b)** Two-qubit gate: A gate can be applied between neighboring qubits but restricts the simultaneous execution of other entangling gates on nearby atoms. **(c)** Long-Range interactions: For gates with larger interaction radii, the restriction zones also expand, resulting in more restricted atoms. **(d)** CCZ gate with a line arrangement of the qubits. According to [11] it is sufficient if the central atom interacts with both the outer qubits. **(e)** CCCZ gate **(f)** Shuttling operation: AOD (blue) enable the movement of atoms within the same column or row. **(g)** Additional NA capabilities, useful for future fault-tolerant computations. Not relevant for this work [18].

5 Compilation in NAQC

Compilation is a process of translating a high-level, abstract description of a quantum algorithm into a low-level representation of operations suitable for hardware execution. To achieve it, this problem is divided into multiple subproblems, employing different layers of software each for specific subproblems, well-known as compilation toolchain. Moreover, compilation should be familiar with computational capabilities of target hardware. As described in 4 each compilation step for NAQC should consider further hardware constraints.

Described by [22, 18]; For example, that together with SWAP or instead of SWAP-based losing of connectivity problem there is also possible to shuttle qubit physical, compiler should be able to calculate which way will be the most efficient in terms of speed, fidelity, calculation time in the specific situation.

Hence, compiler should consider different execution times of gates, their fidelities, and also an available set of gates on current machine.

Moreover, compiler should consider idle time of qubits and calculate corresponding impact from coherence time. Thus, it should now which operations could be executed parallel and what properties do AOD and SLM have.

Nevertheless, compiler should consider possible different grid architectures as described in 4 to take advantage from different lattice realizations. Consequently, as described in 4 interaction is not restricted only for interacting qubits but decays with distance, following an inverse cubic law.

Also, compiler should consider different interaction radius and Rydberg blockade radius, and schedule next step according to constraints that go from previous steps and architecture, to improve overall quality of compilation.

5.1 Compilation Steps

Consider 5.1.

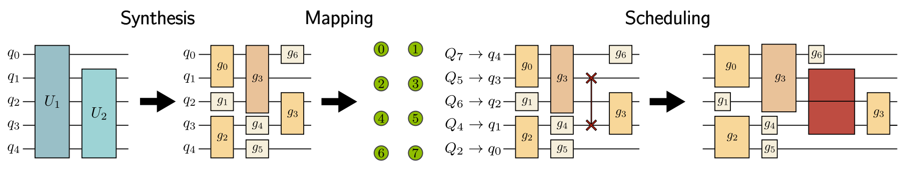


Figure 5.1: **Illustration of the three steps for platform-dependent compilation.** In the **synthesis** step, general operations and unitaries are decomposed into the native gate set Σ_{native}. During the **mapping** step, the circuit qubits q_i are assigned to physical hardware qubits Q_i, and necessary SWAP or MOVE operations are introduced to satisfy connectivity constraints. Finally, in the **scheduling** step, gate times and restrictions on parallelism are considered. In practice, these steps are often performed simultaneously as a single step rather than sequentially. [18]

5.2 Considered Toolchains

In this work three Compiling Toolchains for NAQC are considered, since they take a QASM circuit as import and are open-source python projects [22, 19, 7]. This chapter will cover algorithms that are used by these three, describe a workflow, what tools each toolchain uses, and what is considered by compilation. Nevertheless, the result of benchmarking will be evaluated in the next chapter.

5.2.1 HybridMapper MQT

HM, a tool from MQT, stands out from the previous works, because it uses a hybrid compilation approach to map a circuit and considers together SWAP gates and shuttling to explore the potential advantage of leveraging gate-based SWAP insertions and shuttling-based atom rearrangements [19]. When other works individually only considered it separately [22, 7, 12]. In particular, this is only a mapping and scheduling stage of computation, without synthese and optimization steps. The process takes a QASM circuit as input and uses only gates that defined with architecture along with their times, and fidelities. HM does not optimize circuit or change something in there, it finds places where interacting conditions are violated and solves it with logical or physical move according to internal cost function [19]. In advance, skipping synthese step gives advantages, since other considered toolsets always try to transpile input circuit into Pauli Gates and CZ gate, without consideration of possible multi-qubit CZ gates [19].

The main idea of algorithm is to use two-capability-specific heuristic cost functions, which specially made for fast evaluation and consider additional architecture information such as number of AOD or SLM traps, distance between them, interaction radius to improve parallelism by using commutation rules and look-ahead functionality [19]. Consider for accuracy 5.2.

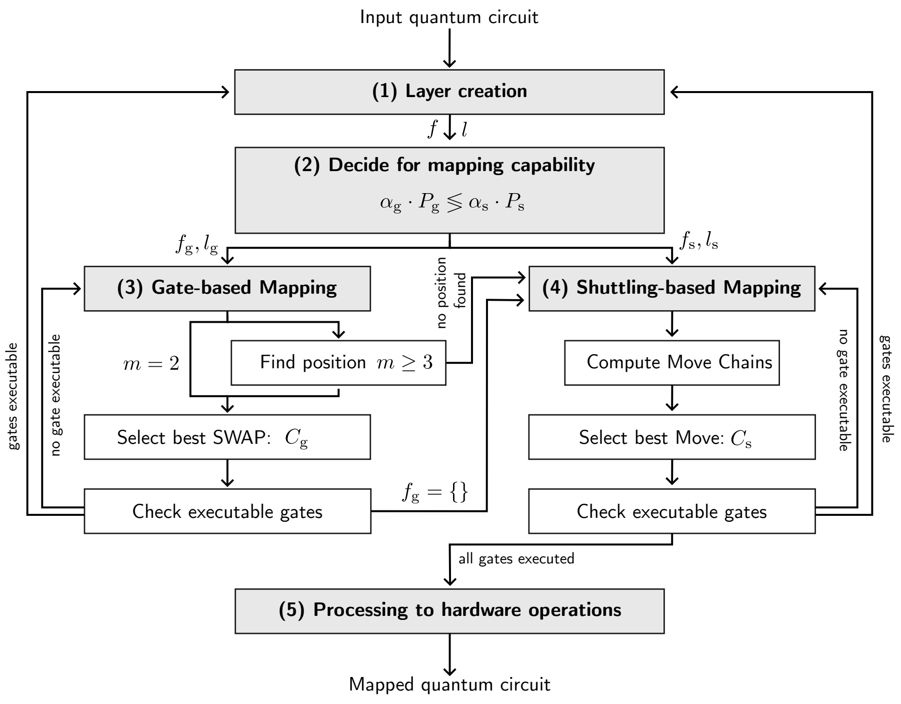
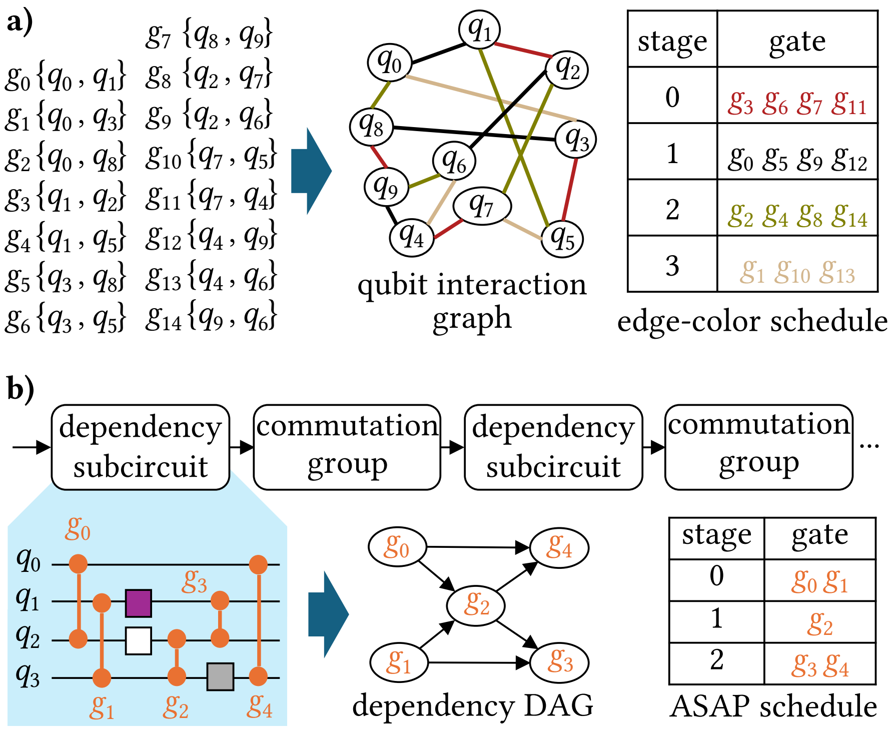


Figure 5.2: Resulting hybrid mapping process [19]

5.2.2 Enola

Enola a Tool from UCLA-VAST based on OLSQ-DPQA uses different approach [22, 21]. It divides mapping and scheduling steps of compilation into scheduling, placement and routing steps to achieve excellent fidelity.

Scheduling step is implemented thorough Edge Coloring problem of commutation group (a group consisting of commutable two-qubit gates that can be executed in any order) graph, where vertices are qubits and edges are two-qubit gates. This problem then is solved using Misra-Gries algorithm. For more generic circuits Enola can use the dependency DAG (directed acyclic graph) for the two-qubit gates in a generic circuit. In this case, the scheduling problem is straightforward: the optimal number of stages is the critical path in the DAG and ASAP (as soon as possible) scheduling can achieve optimality [22]. Visualization 5.3



stage	gate
0	g_3 g_6 g_7 g_{11}
1	g_0 g_5 g_9 g_{12}
2	g_2 g_4 g_8 g_{14}
3	g_1 g_{10} g_{13}

stage	gate
0	g_0 g_1
1	g_2
2	g_3 g_4

Figure 5.3: Scheduling in Enola. **(a)** Scheduling a commutation group of two-qubit gates with edge coloring. **(b)** Generic circuits can be divided to dependency subcircuits and commutation groups. Dependency subcircuits are scheduled ASAP [22].

Placement step is implemented through fast simulated annealing algorithm, qubits are mapped to interaction sited and the two-qubit gates at each Rydberg stage are like 2-pin nets in conventional circuit placement. Hence, the goal is to minimize total "wire-length". Fast simulated annealing has three-stages to explore possible states. At the first stage random search to explore a large solution space is used, the so-called temperature is high, thus a probability of bad solution is high. Then, the second stage is a pseudo-greedy local search. The last stage is a hill-climbing search where the temperature increases again to escape from local minima [22]. Visualization 5.4.

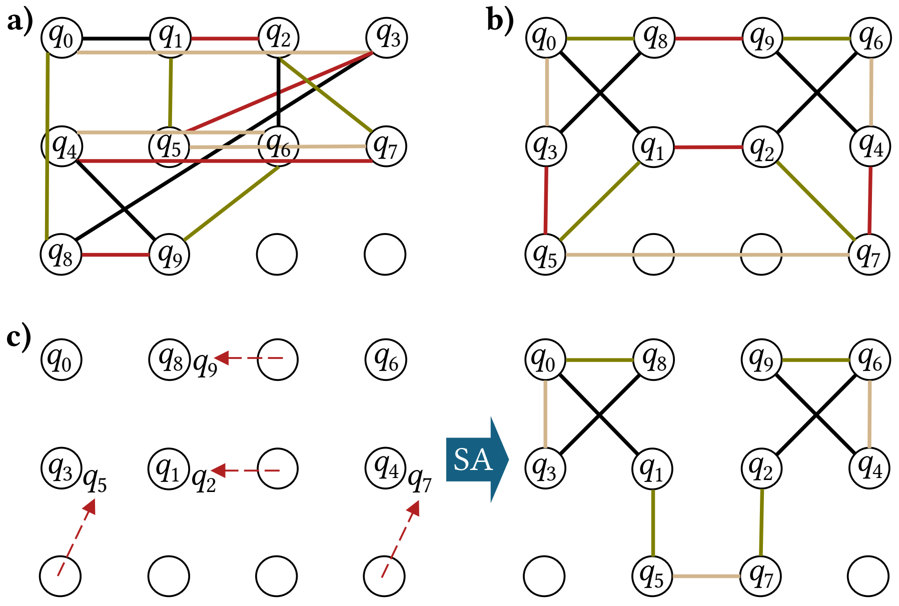


Figure 5.4: Placement in Enola. **(a)** Trivial placement from left to right, from top to bottom. **(b)** Placement with gate distance optimized by simulated annealing. **(c)** Dynamic placement: after a Rydberg stage (red) is executed (left), run simulated annealing on moved qubits for a new placement (right). [22]

Routing step is used to parallelize the AOD movements, and not to violate fundamental rules of AOD such as *the order of its columns cannot change, nor can the order of rows*. Then the independent set of vertices is made of possible moves. This set will be solved then with maximum independent set solver [22]. Visualization 5.5.

Additionally, Enola transpiles input circuit into fixed gate set from Pauli Gates and CZ, without consideration of a big advantage of NAQC that allows to implement different gates. Nevertheless, it considers interaction radius, Rydberg blockade radius, durations and fidelities of gates, physical placement of SLM and AOD traps. An important note is that Enola does not consider number of AOD, but make an independent set of gates

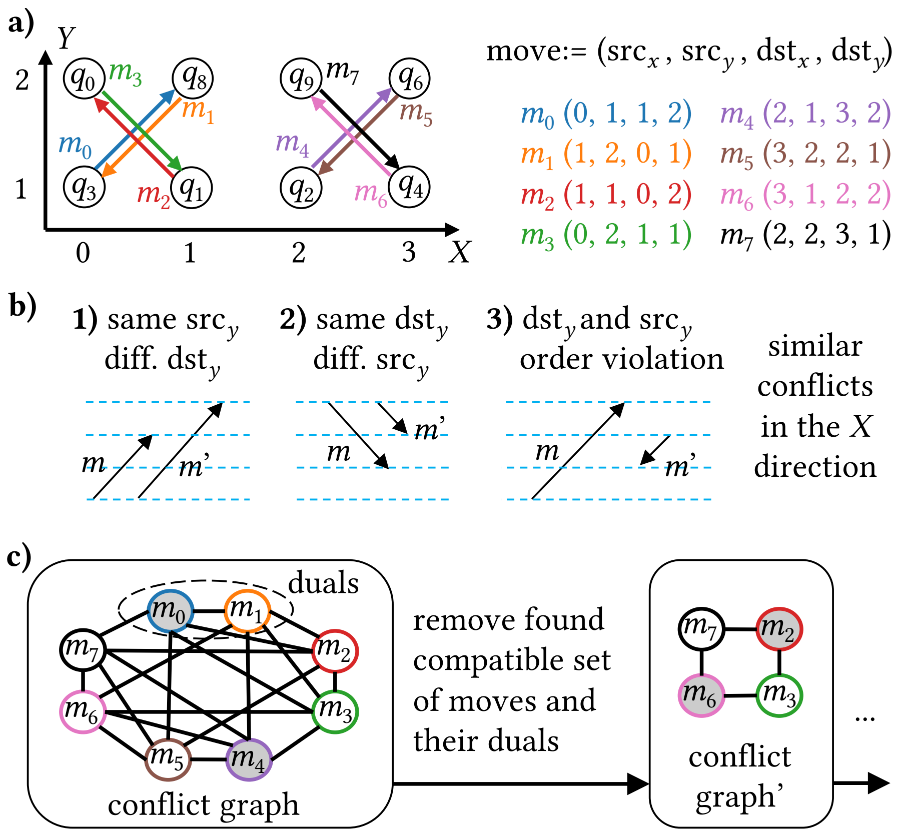


Figure 5.5: Routing in Enola. **(a)** Definition of a move as a 4-tuple. **(b)** Conflicts between two moves. **(c)** Compatible moves are independent sets (IS) in the conflict graph (filled vertices). After finding an IS, delete the moves and their duals from the graph. The process continues until no moves are left [22].

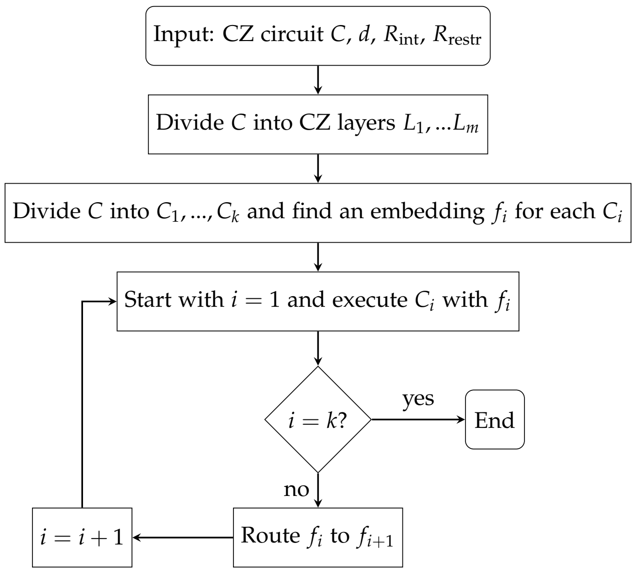

Input: CZ circuit C, d, R_{int}, R_{restr}
Divide C into CZ layers $L_1, ...L_m$
Divide C into $C_1, ..., C_k$ and find an embedding f_i for each C_i
Start with $i = 1$ and execute C_i with f_i
$i = k$?
yes
End
no
Route f_i to f_{i+1}
$i = i + 1$

Figure 5.6: The flowchart of DasAtom [7].

so-called stages where each pair of gates could be executed without affecting other qubits in corresponding stage [22].

5.2.3 DasAtom

DasAtom was made as an improvement of Enola and Tetris and used the weakest aspects of each to strengthen them and made itself. Enola leverages atom shuttling to adapt qubit mappings dynamically, but cannot take advantage of long-range interactions and Tetris' main idea is to leverage the long-range interactions of NAQC to achieve denser qubit connectivity [12, 22, 7].

The main idea of DasAtom is a slightly changed Divide-and-conquer (DAC) approach to divide circuit into subcircuits. It was inspired from different DAC adaptations from [20, 25, 8] Then it assigns an optimal qubit mapping for each subcircuit, and then shuttles atoms to smoothly transition between mappings. This approach should improve overall fidelity and efficiency [7]. Consider 5.6

DasAtom states about 415.8 times higher fidelity comparing to Enola in Quantum Fourier transform (QFT)30. This statement seems to be only partially correct. [7]

6 Evaluation of Framework

For benchmarking of discussed in 5 compilation tool chains a python project was introduced [9]. The initial objective was to compare these three tool chains with architectures as similar as possible.

6.1 Implementation and testing nuances

For those purposes the following program structure were made 6.1. As Test Cases several QFT circuits are used. Since all considered compilers are not actually a compiler

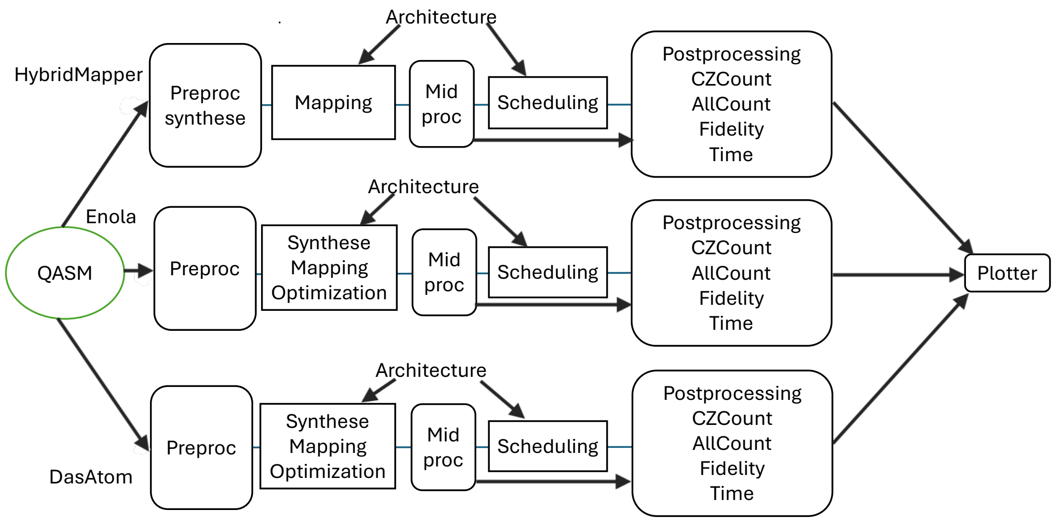


Figure 6.1: Workflow of benchmarking script

in the full sense, but rather tool chains with different tools a multitude non-trivial architecture adaptations, pre-processing, mid-processing and post-processing steps were required.

6.2 First Evaluation

Here are main architecture parameters for first evaluation 6.1. Then they were adapted as similar as possible to pass into each compiler's own input form, as a result, small inaccuracies may occur.

Table 6.1: Architecture parameters for first run

Parameter	Value
Interaction Radius	2
Rydberg Blockade	2
Two Qubit Time	0.36
One Qubit Time	0.36
Two Qubit Fidelity	0.9999
One Qubit Fidelity	0.9999
Coherence Time μs	1500000
AOD Activate Time μs	20
Move Fidelity	0.9999
Move Speed μm/μs	0.55
SLM AOD separation	2

The following results were obtained 6.2, 6.3, 6.4. Nevertheless, due to very long compilation time Enola was tested separately in QFT30 6.2.

6.2.1 Result Interpretation

Firstly, one can observe 6.2 that number of used gates by HM with Shuttling-based strategy is the lowest, when other tools require significantly more gates. It is a case, because HM in shuttling mode only make moves and does not change a circuit, when other actively change input circuit and HM in SWAP-based mode uses a CZ with Hadamard gates to solve a coupling problem [19, 9].

Secondly, one can see in 6.2, in 6.2, and in 6.3 that Enola uses considerable fewer gates one-qubit Gates than DasAtom. But when one sees an output of compilation, then a fidelity of one-qubit gates is equal to 1.0 since both Enola and DasAtom do not

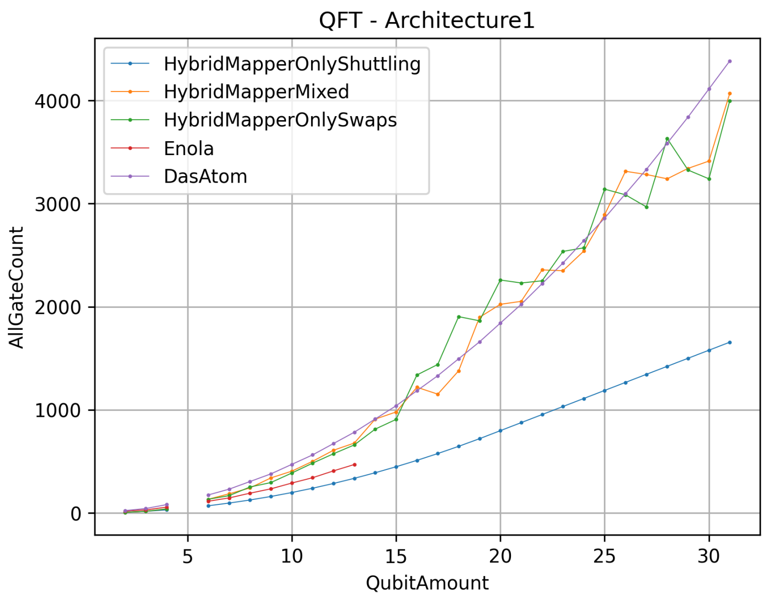


Figure 6.2: Number of All Gates in compiled Circuit in first architecture

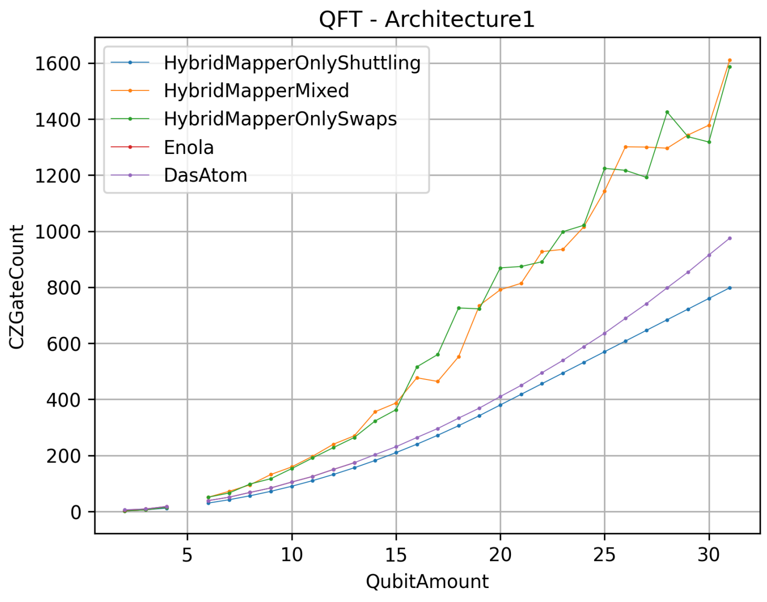


Figure 6.3: Number of CZ Gates in compiled Circuit in first architecture

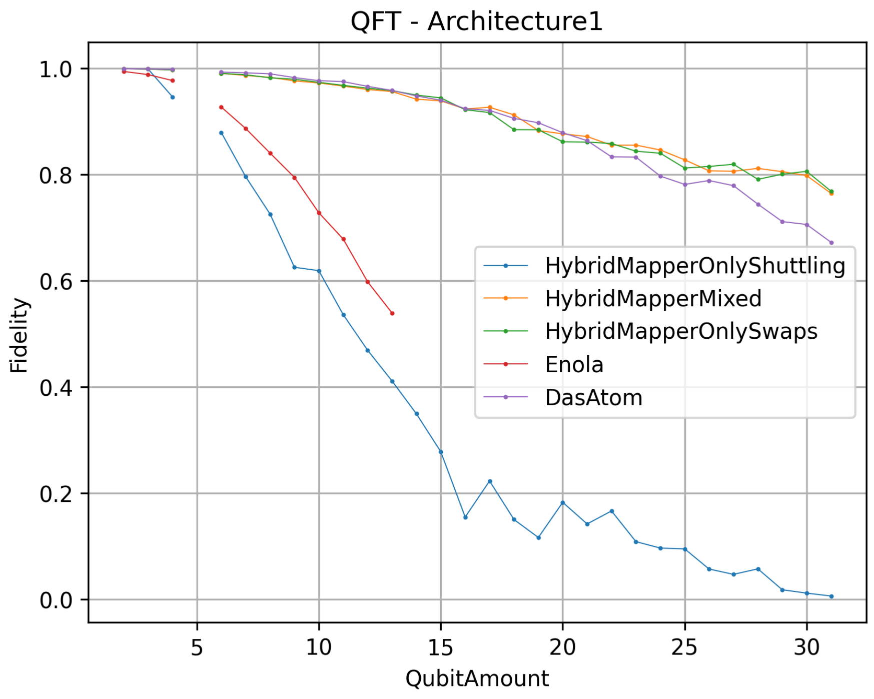


Figure 6.4: Fidelity in first Architecture

Table 6.2: Enola DasAtom First Run QFT30

Output	Enola QFT30	DasAtom QFT30
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

consider impact of one-qubit gates onto fidelity. Taking into account that DasAtom states 415x times more fidelity than Enola, but in same time uses more and not consider one-qubit gates. This may result in a comparison that favors one toolchain over the others. For example, for fidelity of one-qubit gates equal 0.9999:

$$0.9999^{4111-2370} \approx 0.84$$

What would bring a valuable impact on overall fidelity.

Thirdly consider 6.4 and 6.2, QFT is not a very large circuit, and parameters for physical AOD and SLM grid were considerable similar. It is worth investigating the reasons behind this difference in move fidelity and coherence fidelity between Enola and DasAtom. Recall that DasAtom states a significant outperform of Enola and according to results of first run fidelity difference were around 700x times, what is similar in inaccuracy with 414x times [7].

Fourthly, it is notable that optimization of DasAtom achieve the same fidelity as Swap Based HM, when DasAtom uses only shuttling and HM with shuttling is substantially worse.

6.3 Justification and correction of differences

To find out the reason for such a strong difference a source codes of corresponding gits of Enola [22] and DasAtom [7] were investigated [9].

6.3.1 Single Qubit Fidelity

Recall that DasAtom and Enola does not consider one-qubit gates into fidelity calculation. Nevertheless, Enola introduces one-qubit fidelity calculation in its paper, however, in the actual source code this value is hardcoded as 1.0 and remains unused [9, 22]. Therefore, such functionality to both compilers was added according to well-known formula:

$$fidelity_{1Q}^{N_{1Q}}$$

6.3.2 Investigating Different Coherence Fidelity

In Enola according to source codes, and it's paper following formula for fidelity coherence is used:

$$\prod_{q \in Q} \left(1 - \frac{t_q}{T_2}\right)$$

It is a first order Taylor expansion and enough precise, but better to use an exponential variation which is used in DasAtom:

$$e^{-t/T_2}$$

After applying modifications, Enola calculates coherence fidelity with:

$$\prod_{q \in Q} \left(e^{-t_q/T_2} \right)$$

6.3.3 Investigating Different Move Idle Time for Coherence

During improving of Enola behavior, was noticed, that an idle time t is significantly different for Enola compared to the other two. The search revealed the reason for this difference; DasAtom and HM use simple linear formula to calculate time for movement:

$$distance/speed$$

Nevertheless, Enola used an approach described by Dolev Bluvstein to calculate move time:

$$200\sqrt{\frac{distance}{110}}$$

This approach does not consider different possible architectures and therefore speed and distances. It was created to show a non-linear dependency between time and distance e.g. due to acceleration and slowdown. But when distance is not equal to 110 then a difference between resulting times of approaches grows drastically. Hence, for honest testing Enola will use also linear approach.

6.3.4 Investigating Different Move Distances

The following issues were noticed during repair of move time calculation [9]. Move distance was very different between DasAtom and Enola. For example, when average movement on DasAtom was 11 μm, on Enola it was more than 200 μm. That was strange and needed further investigation.

As a result a significant source code error was found. The Enola compiler does not consider architecture parameters on mapping and routing steps, only on scheduling. Therefore, from the outside it looked like there was some kind of reaction to the architecture changes. But in mapping and routing step the architecture parameters were defined as global variables, when a Set function does not consider those as global, but creates local variables with same name. This mistake was fixed and changes could be evaluated again.

6.4 Second Evaluation

Here are slightly revised new architecture parameters based on increased experience working with these toolchains. Changed was a AOD activate time from 20 μs to 0.55 μs due to the diverse utilizations of this parameter, and therefore different impact on similar AOD sequences. Moreover, fidelity of two-qubit gates were downgraded from 0.9999 to 0.9996 due to added consideration of one-qubit gates with fidelity 0.9999, and obvious that two-qubit gates should have then lower fidelity.

Table 6.3: Architecture parameters for second run

Parameter	Value
Interaction Radius	2
Rydberg Blockade	2
Two Qubit Time	0.36
One Qubit Time	0.36
Two Qubit Fidelity	0.9996
One Qubit Fidelity	0.9999
Coherence Time μs	1500000
AOD Activate Time μs	0.55
Move Fidelity	0.9999
Move Speed μm/μs	0.55
SLM AOD separation	2

The following results were obtained 6.5, 6.6, 6.7.

6.4.1 Result Interpretation

Firstly, one can observe in 6.7 that difference between DasAtom and Enola for QFT30 is only about 5.5 times higher, and an exponentially increasing gap is not observed. Moreover, Enola consider a lot more fidelity affecting factors that were not considered here and not investigated, but have a considerable effect on fidelity. Therefore, it is likely that, under fair consideration of all co-factors the result difference will be slightly less than 5.5 times.

Secondly, 6.7 confirms that Enola is a bit better than shuttling based HM, due to missing optimization step in HM and DasAtom could achieve a SWAP level fidelity by using only shuttling and sometimes outperform HM with enabled SWAP. Nevertheless, one should not forget that an input QFT is a optimized already by input and perhaps on random circuit the results will be different.

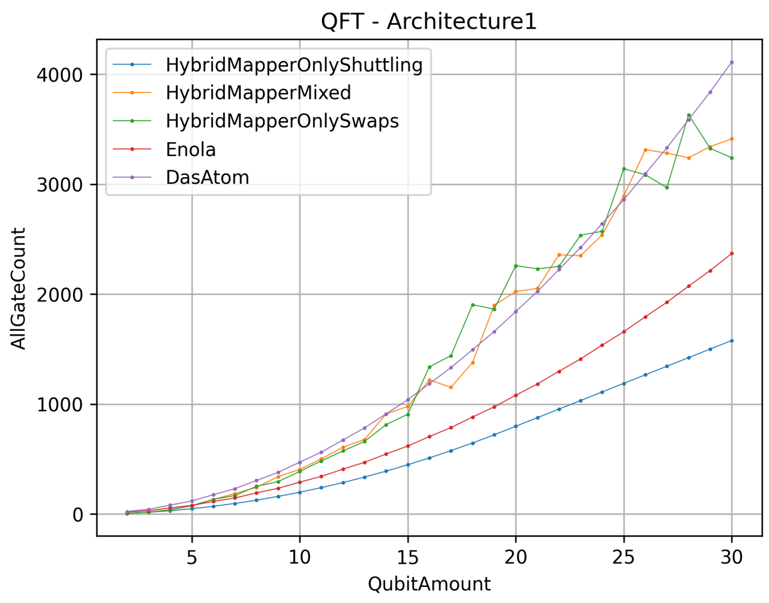


Figure 6.5: Number of All Gates in compiled Circuit in second architecture

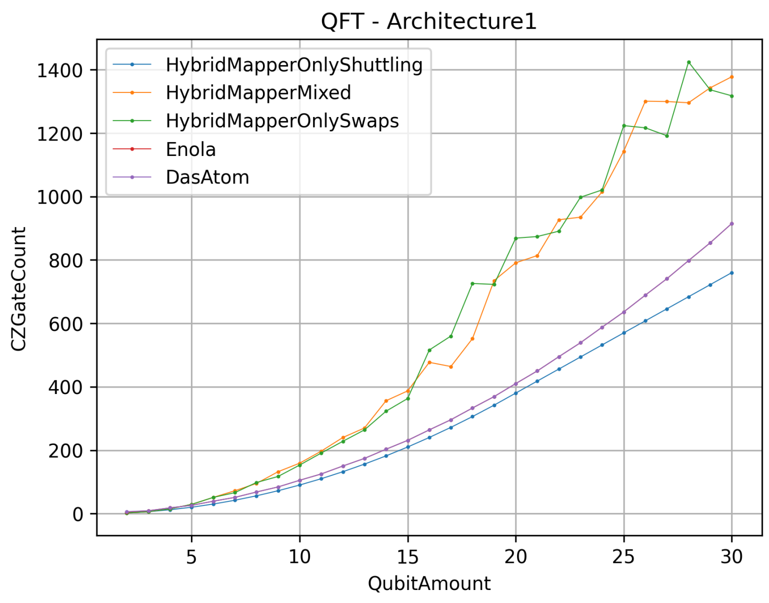


Figure 6.6: Number of CZ Gates in compiled Circuit in second architecture

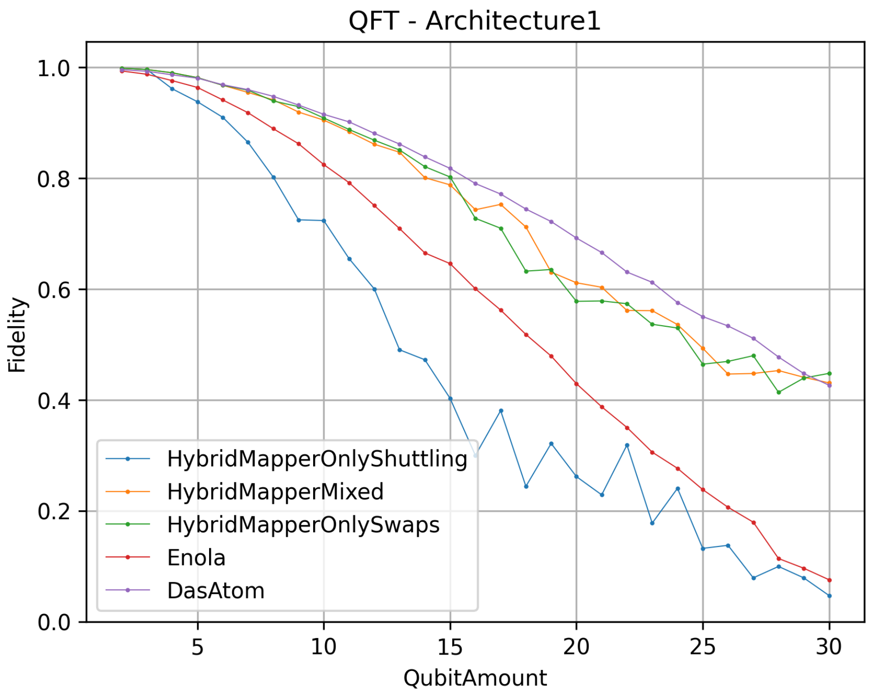


Figure 6.7: Fidelity in second Architecture

7 Future Work

Taken together, much work remains to be done. Firstly, such as adding new compiler toolchains for comparison and developing a universal framework to manage input, and output results, and moreover to fair calculate and interpret statistics. Or something of greater depth, such as consideration of all possible parameters that take place for example in Enola but do not in other, and due to not very big impact on calculations, frequently it is not being considered. Moreover, different circuits could be used, with different level of start optimizations to see actually performance of a tool chain.

Very valuable scientific contribution would be to test compiler tool chain on a real hardware. This would allow for fairer comparisons and a more accurate interpretation towards a potential unified framework. But is it a bit difficult due to the lack of public available NAQC. Particular ones of them do not expose in API a possibility to shuttle qubit in runtime, such as Aquila from Amazon. Other, work on a very low level of analog pulses, such as Pasqal Pulser, what creates more constraints onto testing of High-level toolchains.

8 Conclusion

NAQC represent a new hardware architecture with exclusive combination of capability and constraints. This architecture has significant perspectives in world of quantum computations. Nevertheless, to subjugate all the power of it a good software solution must be produced.

This work was focused on benchmarking of existing compilation toolchains and the most important thing is an honest comparison. What was fullified in this paper.

Firstly, a test setup was created, to bring all tool chains into similar conditions, and first testing was performed.

Secondly, results were interpreted, and suspicious things were investigates and repaired.

Thirdly, the new testing was performed, that has shown a worthy results such as finding that difference is actually not a 400x times and exponential grow as DasAtom [7] states, but either 5.5x times between Enola and DasAtom [9].

Abbreviations

DPQA Dynamically Field-Programmable Qubit Arrays

NAQC Neutral Atom Quantum Computer

QC Quantum Computer

NISQ Noisy Intermediate Scale Quantum Computing Era

FTQC Fault-Tolerant Quantum Computing Era

SLM Spatial Light Modulator

AOD Acousto-Optic Deflector

MQT Munich Quantum Valey

UCLA-VAST UCLA-VAST Lab

QFT Quantum Fourier transform

DAC Divide-and-conquer

HM HybridMapper

List of Figures

List of Tables

Bibliography

[1] N. Barde, D. THAKUR, P. Bardapurkar, and S. Dalvi. "Consequences and Limitations of Conventional Computers and their Solutions through Quantum Computers." In: *Leonardo Electronic Journal of Practices and Technologies* 10 (July 2011), pp. 161–171.

[2] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. "Elementary gates for quantum computation." In: *Physical Review A* 52.5 (Nov. 1995), pp. 3457–3467. ISSN: 1094-1622. DOI: 10.1103/physreva.52.3457. URL: http://dx.doi.org/10.1103/PhysRevA.52.3457.

[3] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage. "Trapped-ion quantum computing: Progress and challenges." In: *Applied Physics Reviews* 6.2 (May 2019). ISSN: 1931-9401. DOI: 10.1063/1.5088164. URL: http://dx.doi.org/10.1063/1.5088164.

[4] D. Gao, Y. Li, S. Ying, and S. Li. *Optimal Compilation Strategies for QFT Circuits in Neutral-Atom Quantum Computing*. 2025. arXiv: 2506.15116 [quant-ph]. URL: https://arxiv.org/abs/2506.15116.

[5] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak. "Quantum computing with neutral atoms." In: *Quantum* 4 (Sept. 2020), p. 327. ISSN: 2521-327X. DOI: 10.22331/q-2020-09-21-327. URL: http://dx.doi.org/10.22331/q-2020-09-21-327.

[6] H.-L. Huang, D. Wu, D. Fan, and X. Zhu. "Superconducting quantum computing: a review." In: *Science China Information Sciences* 63.8 (July 2020). ISSN: 1869-1919. DOI: 10.1007/s11432-020-2881-9. URL: http://dx.doi.org/10.1007/s11432-020-2881-9.

[7] Y. Huang, D. Gao, S. Ying, and S. Li. *DasAtom: A Divide-and-Shuttle Atom Approach to Quantum Circuit Transformation*. 2025. arXiv: 2409.03185 [quant-ph]. URL: https://arxiv.org/abs/2409.03185.

[8] Y. Huang, X. Zhou, F. Meng, and S. Li. *Qubit Mapping: The Adaptive Divide-and-Conquer Approach*. 2024. arXiv: 2409.04752 [quant-ph]. URL: https://arxiv.org/abs/2409.04752.

[9] E. Khusainov. *Tooling and Benchmarking of a Hardware-Agnostic Compilation Toolchain For Neutral-Atom Quantum Computers*. `https://github.com/i2-tum/Bachelor-Thesis-Emil-Khusainov`. GitHub repository. 2025. (Visited on 06/28/2025).

[10] V. Lahtinen and J. Pachos. "A Short Introduction to Topological Quantum Computation." In: *SciPost Physics* 3.3 (Sept. 2017). ISSN: 2542-4653. DOI: `10.21468/scipostphys.3.3.021`. URL: `http://dx.doi.org/10.21468/SciPostPhys.3.3.021`.

[11] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler, and M. D. Lukin. "Parallel Implementation of High-Fidelity Multiqubit Gates with Neutral Atoms." In: *Physical Review Letters* 123.17 (Oct. 2019). ISSN: 1079-7114. DOI: `10.1103/physrevlett.123.170503`. URL: `http://dx.doi.org/10.1103/PhysRevLett.123.170503`.

[12] Y. Li, Y. Zhang, M. Chen, X. Li, and P. Xu. "Timing-Aware Qubit Mapping and Gate Scheduling Adapted to Neutral Atom Quantum Computing." In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 42.11 (2023), pp. 3768–3780. DOI: `10.1109/TCAD.2023.3261244`.

[13] I. L. Markov. "Limits on fundamental limits to computation." In: *Nature* 512.7513 (Aug. 2014), pp. 147–154. ISSN: 1476-4687. DOI: `10.1038/nature13570`. URL: `http://dx.doi.org/10.1038/nature13570`.

[14] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[15] J. Preskill. "Quantum Computing in the NISQ era and beyond." In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: `10.22331/q-2018-08-06-79`. URL: `http://dx.doi.org/10.22331/q-2018-08-06-79`.

[16] QuEra Computing Inc. *Understanding Fault-Tolerant Quantum Computing*. Accessed: 2025-07-10. Nov. 2023. URL: `https://www.quera.com/blog-posts/understanding-fault-tolerant-quantum-computing`.

[17] J. Romero and G. Milburn. *Photonic Quantum Computing*. 2024. arXiv: `2404.03367 [quant-ph]`. URL: `https://arxiv.org/abs/2404.03367`.

[18] L. Schmid, D. F. Locher, M. Rispler, S. Blatt, J. Zeiher, M. Müller, and R. Wille. "Computational capabilities and compiler development for neutral atom quantum processors—connecting tool developers and hardware experts." In: *Quantum Science and Technology* 9.3 (Apr. 2024), p. 033001. ISSN: 2058-9565. DOI: `10.1088/2058-9565/ad33ac`. URL: `http://dx.doi.org/10.1088/2058-9565/ad33ac`.

[19] L. Schmid, S. Park, S. Kang, and R. Wille. *Hybrid Circuit Mapping: Leveraging the Full Spectrum of Computational Capabilities of Neutral Atom Quantum Computers*. 2023. arXiv: 2311.14164 [quant-ph]. URL: https://arxiv.org/abs/2311.14164.

[20] M. Y. Siraichi, V. F. D. Santos, C. Collange, and F. M. Quintão Pereira. "Qubit allocation as a combination of subgraph isomorphism and token swapping." In: *OOPSLA*. Vol. 3. Athens, Greece, Oct. 2019, pp. 1–29. DOI: 10.1145/3360546. URL: https://inria.hal.science/hal-02316820.

[21] D. B. Tan, D. Bluvstein, M. D. Lukin, and J. Cong. "Compiling Quantum Circuits for Dynamically Field-Programmable Neutral Atoms Array Processors." In: *Quantum* 8 (Mar. 2024), p. 1281. ISSN: 2521-327X. DOI: 10.22331/q-2024-03-14-1281. URL: http://dx.doi.org/10.22331/q-2024-03-14-1281.

[22] D. B. Tan, W.-H. Lin, and J. Cong. "Compilation for Dynamically Field-Programmable Qubit Arrays with Efficient and Provably Near-Optimal Scheduling." In: *Proceedings of the 30th Asia and South Pacific Design Automation Conference*. ASPDAC '25. ACM, Jan. 2025, pp. 921–929. DOI: 10.1145/3658617.3697778. URL: http://dx.doi.org/10.1145/3658617.3697778.

[23] K. Wintersperger, F. Dommert, T. Ehmer, A. Hoursanov, J. Klepsch, W. Mauerer, G. Reuber, T. Strohm, M. Yin, and S. Luber. "Neutral atom quantum computing hardware: performance and end-user perspective." In: *EPJ Quantum Technology* 10.1 (Aug. 2023). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-023-00190-1. URL: http://dx.doi.org/10.1140/epjqt/s40507-023-00190-1.

[24] P. W. Wondra. "Neutral Atom Based Multi-Qubit-Gate Synthesis." en. MA thesis. TU München, 2024, p. 54.

[25] T.-A. Wu, Y.-J. Jiang, and S.-Y. Fang. "A Robust Quantum Layout Synthesis Algorithm with a Qubit Mapping Checker." In: *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*. ICCAD '22. San Diego, California: Association for Computing Machinery, 2022. ISBN: 9781450392174. DOI: 10.1145/3508352.3549394. URL: https://doi.org/10.1145/3508352.3549394.

[26] S.-Q. Zhang, J.-S. Hong, Y. Xue, X.-J. Luo, L.-W. Yu, X.-J. Liu, and X. Liu. "Ancilla-free scheme of deterministic topological quantum gates for Majorana qubits." In: *Physical Review B* 109.16 (Apr. 2024). ISSN: 2469-9969. DOI: 10.1103/physrevb.109.165302. URL: http://dx.doi.org/10.1103/PhysRevB.109.165302.