SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - COMPUTER SCIENCE

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Neutral Atom Based Multi-Qubit-Gate Synthesis

Philipp Wolfgang Wondra

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY - COMPUTER SCIENCE

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Neutral Atom Based Multi-Qubit-Gate Synthesis

Multiqubit Gatter Synthese für Schaltkreise auf Neutral-Atom-Quantencomputern

Author:	Philipp Wolfgang Wondra
Supervisor:	Prof. Dr. Christian Mendl
Advisor:	M.Sc. Yanbin Chen
Submission Date:	8.10.2024

Acknowledgments

I would like to express my gratitude to my supervisor, Yanbin Chen. His regular and constructive feedback has greatly assisted me in the development and direction of my work. Particularly helpful were the theoretical discussions during our meetings, which focused on the significance of specific parameters and their underlying causes.

Abstract

Neutral Atom Quantum Computers are a novel architecture within the field of quantum computing. These systems are based on neutral Rubidium, Strontium, or Caesium atoms, which are fixed in a specific arrangement using optical dipole traps [1]. With the help of laser pulses, the individual atoms can be precisely transferred to an electronically excited state. These states are known as Rydberg states and are characterized by a strong polarization, which allows them to interact with other atoms. This enables the realization of arbitrary computational operations and gates between two or more qubits. The ability to natively implement gates with more than two qubits is one of the major advantages of NAQC, as a three-qubit gate can often replace multiple two-qubit gates. This can significantly reduce both the circuit depth and the number of required pulses. Since each gate introduces an error rate into the system, this optimization also has a positive effect on the overall size of the achievable circuits and the expected error rate. The aim of this work is to quantify the performance gains from using multi-qubit gates. Furthermore, selected algorithms will be optimized using four-qubit gate synthesis.

Contents

1. Introduction

"Nature isn't classical, dammit, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem because it doesn't look so easy" [2]. With these words, Feynman described an entirely new type of computing machines designed specifically to simulate quantum mechanical systems and their properties [3]. In addition to their ability to accurately simulate physical systems, quantum computers offer several other advantages. These include the ability to factorize prime numbers in polynomial time [4] and to accelerate search algorithms within unsorted databases [5]. Furthermore, quantum computers enable the reliable transmission of information, known as quantum cryptography [6]. The advantages of quantum computers over classical computers are mainly based on two quantum mechanical principles: the superposition of individual qubits and their entanglement.

While the fundamental principles of quantum mechanics are universal, there is no single universal architecture for quantum computers. Instead, there are many possible architectures, each with its own advantages and disadvantages [7]. Among the most prominent architectures are superconducting qubits, nuclear magnetic resonance devices, and trapped ion quantum computers [8, 9, 10]. One promising architecture is neutral atom quantum computers (NAQC). One of the central advantages of this approach is the inherent scalability of neutral atom quantum computers [11]. This scalability is essential because quantum computers differ drastically from classical computers in terms of the resources required for error correction schemes. While classical computers need a few tens of percent of their resource capacity, quantum computers implement single logical qubits using an ensemble of up to 100 physical qubits [12].

The implementation of the neutral atom architecture using Rubidium-87 atoms has the additional advantage of offering various states. These include hyperfine ground states, which have very long coherence times and do not interact with other atoms, allowing qubits in this state to function as memory. In contrast, qubits in the Rydberg state can form strong interactions but have a relatively short lifespan. The ability to implement both states in a single atom and switch between them using a laser is a significant advantage of the neutral atom platform [13]. The aforementioned Rydberg states form the basis for implementing gates between non-adjacent atoms [14]. The use of such interactions results in reduced communication overhead, overall gate count, and depth for compiled programs. However, when using gates between atoms at greater distances, it is important to ensure that the resulting restriction zones do not overlap. These zones expand with distance, limiting the ability to execute parallel gates within the qubit array.

Additionally, the native implementation of multi-qubit operations is another advantageous feature of the neutral atom platform compared to superconducting-based qubit technology [14, 15, 16]. Gates acting on three or more qubits do not need to be decomposed into multiple one- and two-qubit gates and can be executed as is. Since each gate requires a specific number of laser pulses and has an associated error rate, this approach could potentially reduce the pulse count and error rate of a quantum algorithm. To replace a circuit with a circuit where multi-qubit gates are allowed, the computation performed by the latter must be sufficiently similar to the original calculation. The Geyser framework employs the dual-annealing approach for this purpose, a stochastic method developed for parameter calculation in constraint optimization problems [17, 15].

In its current version, the Geyser paper only addresses the optimizations made possible by using the CCZ gate. This does not cover the entire range of optimizations enabled by the ability to natively implement arbitrary multi-qubit gates on the neutral atom architecture. Therefore, this work further explores the optimization of specific quantum algorithms through the use of 4-qubit gates.
To achieve this optimization, the Geyser framework is adapted with respect to the layout of qubit arrays. Additionally, the allocation of computational resources is modified. The impact of a different blocking mechanism is also analyzed. The optimized algorithms are then compared to the original algorithms to examine their similarity. Furthermore, an analysis is conducted to determine which algorithms exhibit the most significant performance gains. Additionally, the computational effort required to calculate a specific similarity of the optimized algorithm is presented. Based on this, an estimate is made of the computational effort needed to optimize arbitrarily large quantum algorithms to any desired accuracy.

In Chapter 2 of this work, we first describe some theoretical concepts of quantum computing. Subsequently, in Chapter 3, we analyze the advantages and disadvantages of the neutral atom architecture and the physical implementation of multi-qubit gates. Chapter 4 details the implementation and adaptation of the Geyser framework. In Chapter 5, the resulting optimized algorithms are evaluated. Finally, Chapter 6 provides a conclusion and discusses potential further improvements.

2. Background

2.1. Quantum Bits

In classical computer science, bits are used to store information. Each bit can be in either state 0 or 1. Information encoded by assembling multiple such bits can then be processed through targeted manipulation of the 0s and 1s. This is achieved using logic gates such as NOT, AND, and OR gates. A characteristic feature of classical computers is that, except for the NOT gate, these gates are not reversible. This becomes evident as it is impossible to deduce the initial states of A and B from the result of A AND B = 0. The initial states could have been 00, 01, or 10. This loss of information leads to the well-known problem of heat dissipation in classical computers [6].
In contrast, quantum computers are based on quantum bits. Like classical bits, qubits also have two fundamental states, which can be represented as $|0\rangle$ and $|1\rangle$ using Dirac notation. However, qubits can also exist in a linear combination of these basic states, a phenomenon known as superposition, which is represented as follows:

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

The factors alpha and beta are complex numbers that represent the amplitudes of the respective base states. The individual amplitudes, and consequently the quantum state as a whole, cannot be measured directly. Instead, a measurement can only determine whether the qubit is in state $|0\rangle$ or $|1\rangle$. The squares of the coefficients, $|\alpha|^2$ and $|\beta|^2$, represent the probabilities that a measurement will yield the respective base state $|0\rangle$ or $|1\rangle$. These coefficients must adhere to Born's rule, which states that the sum of the squares of the amplitudes must equal 1.

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

2.2. Quantum Gates

A key distinction between quantum computers and classical computers is the inherent reversibility required in the former. This is justified by the Schrödinger equation and its principle of unitary time evolution of the wave function, which governs all processes on a quantum computer. Thus, individual gates are always represented by unitary and therefore reversible matrices. Such a matrix has the form $2^n \times 2^n$, where n denotes the number of qubits, and its entries are drawn from $\mathbb{C}$. These gates can be applied to any number of qubits for information processing, altering the current state vector of the entire circuit in the process.

$$U|\psi_1\rangle = |\psi_2\rangle$$

An example is shown in the following equation, where an X-gate is applied to the state vector $|0\rangle$. This gate functions as a NOT gate because it swaps the amplitudes of the basis states of the affected qubit.

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad X|0\rangle = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \cdot 1 + 1 \cdot 0 \\ 1 \cdot 1 + 0 \cdot 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle$$

Other essential matrices for constructing quantum circuits include the Pauli-Z, Hadamard, and U gates. These can be examined in the following representation in that same order. The Z gate leaves the amplitude of the $|0\rangle$ state unchanged and only inverts the sign of the $|1\rangle$ state. The Hadamard matrix creates a superposition if the qubit was initially in a ground state. The U3 gate is a general gate that allows arbitrary rotations of the qubit around its three angles θ, ϕ, and λ [18].

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad U3 = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & e^{i(\phi+\lambda)} \cos(\frac{\theta}{2}) \end{pmatrix}$$

2.3. Multiqubit Gates

Now that we possess all the fundamental tools for the description and manipulation of individual qubits, we need a method to describe systems composed of multiple qubits. The tensor product helps us combine the Hilbert spaces of individual qubits, thereby creating a space in which we can describe the state of the entire ensemble of qubits. This space always contains 2^n vectors for n qubits. The following formula exemplifies the combined Hilbert space for two qubits $|a\rangle$ and $|b\rangle$ [6].

$$|\psi\rangle = |a\rangle|b\rangle = (|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle$$

Based on this representation, we can now construct so-called multiqubit gates, which operate on more than a single qubit. Their distinguishing feature is the capability of applying a change to the state of the target qubits depending on the state of control qubits [19]. An example of this is the CX gate, which inverts the current amplitude of the target qubit if the control qubit has a value of 1 by applying an X gate to it. The corresponding matrix is shown below.

$$\mathrm{CX} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

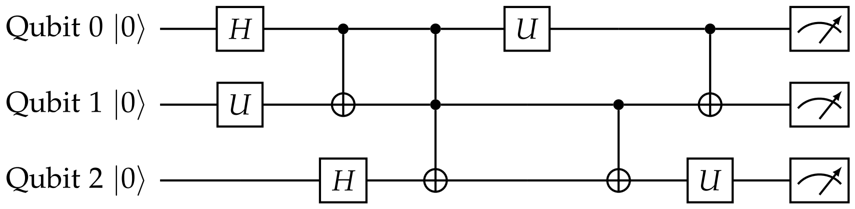
Beyond the CX gate illustrated above, arbitrary multiqubit gates can be constructed using single-qubit-gates. The CCZ gate features two control qubits that, when activated simultaneously, invert the $|1\rangle$ component of the third qubit. The construction of the CCZ gate is demonstrated in the formula below.

$$\mathrm{CCZ} = |0\rangle\langle 0| \otimes I \otimes I + |1\rangle\langle 1| \otimes (|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z)$$

This approach can be extended to realize 4-qubit CCCZ gates and 5-qubit CCCCZ gates. Additionally, Bayram Jumayev presents a universal formula for constructing multiqubit gates with exactly one control qubit [19]. For the remaining qubits of the gate, a computational operation is performed starting with G_1 for the first target qubit. This results in gates such as CXX, CHH, and CXXH, among others.

A quantum circuit is the deliberate execution of several individual gates in a predefined sequence [15]. Since a circuit is therefore only a sequence of several gates on n qubits, it can be described in the same way as a single gate by a unitary matrix. For a circuit with n qubits, this matrix has dimensions $2^n \times 2^n$, with entries drawn from the set of complex numbers $\mathbb{C}$. Two circuits can have different structures yet still possess the same unitary matrix. Since this matrix uniquely describes the entire computation performed by the circuit, the two circuits are equivalent in terms of their computations.

An exemplary circuit can be viewed in the diagram below. On the left side, the qubits are depicted with their respective initial states. From there, the wires, which represent single qubits, extend horizontally to the right [20]. On these wires, individual computational operations can be applied in the form of gates at the desired times. The U gates serve as examples of single-qubit gates. Vertical connection lines between the qubit wires in combination with dots or boxes on the wires represent multi-qubit gates. At the right end, after all necessary computations have been executed, measurement components are placed to capture each single qubit.



3. Neutral-Atom-Architecture

3.1. Computation Cycles on Neutral Atom Quantum Computers

To understand the specific functioning of the neutral atom quantum computer architecture, it is crucial to have an overview of the entire process that any algorithm undergoes on this system. This process can be divided into the sections: register preparation, quantum processing, and register readout [21].

3.1.1. Register Preparation

The register preparation phase encompasses all the steps necessary to enable the quantum computer for performing computational operations. A key distinguishing feature of the NAQC architecture is the use of individual atoms as qubits. The specific choice of atom type significantly influences the entire system. Generally, atoms from the alkali or alkaline earth metal groups are used, as their electronic structure offers advantageous properties for cooling and localization [1]. The most commonly utilized atoms are rubidium (Rb^{87}), cesium (Cs^{133}), ytterbium (Yb^{173}), and strontium (Sr^{87}) [1] [16].

To enable subsequent computations, the selected atoms must first be transferred into a register. Initially, a dilute gas of the chosen atoms is created. This process is carried out at room temperature using an ultra-high vacuum system. Subsequently, a small number of atoms are cooled to approximately 1mK within Magneto-Optical Traps (MOT) [1]. This cooling is achieved through a process called Doppler cooling, which specifically targets the atoms and does not affect the overall room temperature. This effect is induced by lasers tuned to specific frequencies that only interact with the desired atoms.

During the next step, spatial light modulators (SLMs) are employed. These devices focus a laser through high numerical aperture lenses onto a configurable number of points in space. These points, known as optical tweezers, have a diameter of approximately 1 μm, allowing them to only hold a single atom. An individual atom can typically be held in an optical tweezer for an average duration of 10-20 seconds [1]. The array of optical tweezers can be arranged in any 1D, 2D, or 3D configuration. Examples for such arrangements can be seen in figure 3.1.
After successfully completing Steps 1 and 2, each individual optical tweezer contains an atom only 50-60% of the time [1]. Alternatively, this position is empty. Such a pattern, which exhibits irregular vacancies, is unsuitable for computation. Therefore, the empty positions must first be identified. This is achieved using a Charge Coupled Device (CCD) camera

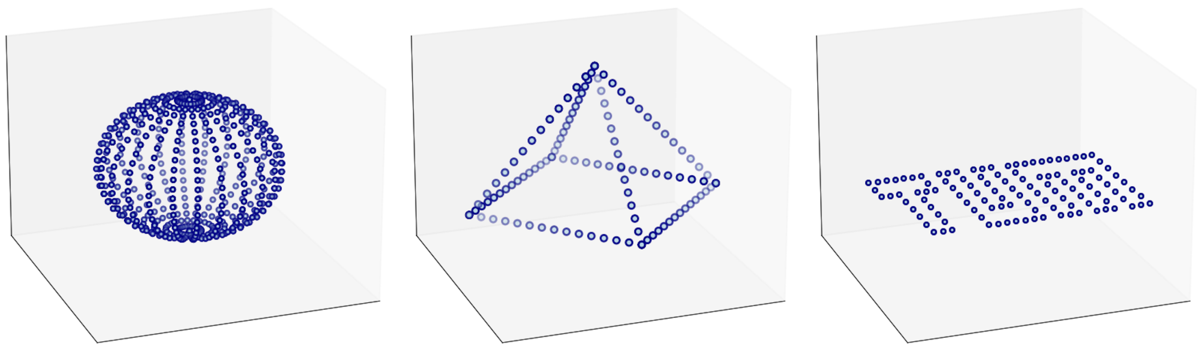


Figure 3.1.: Arbitrary 3D-configurations of Neutral Atom Registers

capable of detecting fluorescence emitted by individual atoms when illuminated with light of the appropriate wavelength. In case the optical tweezer is empty, no fluorescence is emitted, and therefore nothing can be detected.

By knowing the positions of individual atoms, they can now be moved accordingly, allowing the construction of the originally required architecture. However, this success depends on accurately estimating the expected number of vacancies at the outset and preemptively creating a sufficient number of optical tweezers. The movement is achieved using mobile optical tweezers, which can be generated using acousto-optical deflectors (AODs). The programmable motion of these mobile optical tweezers achieves a success rate of over 99%. A final image before computation is captured to ensure the structure and complete occupancy of the arrangement.

3.1.2. Quantum Processing

During the digital quantum processing phase, quantum algorithms are decomposed into a sequence of gates. These gates are then executed on the neutral-atom register prepared in the preceding step. Crucial to this process is the encoding of the qubit ground states $|0\rangle$ and $|1\rangle$ on individual atoms. To enable the computation of extended quantum algorithms, the qubits and more precisely their ground states must exhibit long coherence times, which requires minimal interaction with their environment. In the case of rubidium, two hyperfine ground states of the atom are used to implement $|0\rangle$ and $|1\rangle$ [1]. These are based on the spin of the outermost electron of the rubidium atom. Alternatively, in the case of strontium, the nuclear spin can be used as the state vector for the qubits. However, their targeted manipulation is more challenging than that of electron spins. Nonetheless, nuclear spins exhibit longer decoherence times. An example of the implementation of hyperfine ground states in rubidium atoms can be seen in the graphic below.

In addition to encoding the ground states, the implementation and use of high-fidelity single-qubit gates are essential components of quantum processing. As described in the previous chapter, applying these gates results in a precise rotation of the state vector on the Bloch sphere. This can be visually illustrated with the X-gate and Z-gate on the Bloch sphere shown below.

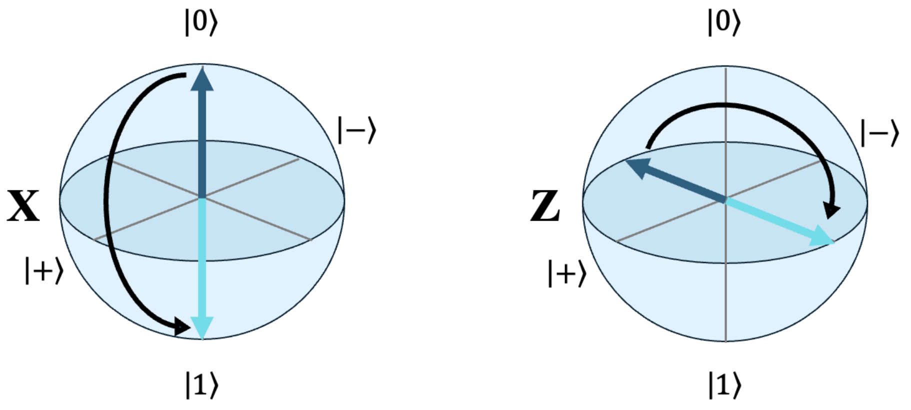


Figure 3.2.: (left) Rotation induced by X-gate (right) Rotation induced by Z-gate

The physical implementation of this transition relies on two lasers. These lasers create a control field that induces so-called Raman transitions. The Raman transition starts with the excitation of the atom by bombarding it with photons, moving it from a hyperfine ground state to a higher-energy intermediate state [22]. This step is called Stokes-Raman scattering. Subsequently, during the Anti-Stokes-Raman scattering process, the atom releases energy by emitting a photon. The key aspect of this process is that the emitted energy is not exactly the same as the initially absorbed energy. As a result, the atom ends up in a different hyperfine ground state. The configuration of specific adjustable parameters enables precise Raman transitions and rotations of the state vector around the X, Y, and Z axes. The key laser parameters are the Rabi frequency Ω, the detuning δ, the relative phase φ, and the duration τ for which the laser is active. From these parameters, the rotation values around the X, Y, and Z axes can be determined using $(\Omega\tau\cos\varphi, \Omega\tau\sin\varphi, \delta\tau)$.

Lastly the execution of arbitrary quantum circuits during quantum processing requires a method for implementing multi-qubit gates. DiVincenzo demonstrated in 1995 that a universal gate set can be achieved using only two-qubit gates [23]. The physical realization of this process in the neutral-atom architecture relies on so-called Rydberg states [21]. These states are characterized by the outermost electron being at a very large distance from the atomic nucleus. This large separation allows for dipole-dipole interactions between multiple atoms, which can be up to 1000 times stronger than in the ground state. The effect of this interaction is repulsive, meaning that only atoms positioned very far apart can simultaneously be in the Rydberg state. When the distance between two atoms is very small, the Rydberg blockade occurs, preventing the later excited atom from reaching the Rydberg state. The critical advancement compared to single-qubit gates is that now the behavior of the second atom can be conditionally dependent on the state of the control atom.

The following section illustrates a specific implementation of the CZ gate. Individual laser pulses are targeted at the atoms and thereby induce a so called π pulse on the atoms. This results in the atoms transitioning to the Rydberg state if and only if they previously were in the $|0\rangle$ state. The choice of the initial ground state from which the Rydberg state is accessed is arbitrary and adjustable. In this CZ gate, the first laser pulse is applied to the control qubit. If the control qubit is in the $|1\rangle$ state, the pulse is off-resonant and does not induce a Rydberg blockade. In the second step, a 2π pulse is applied to the target qubit, which is achieved by sequentially applying two π pulses. The target atom can transition to the Rydberg state only if it was initially in the $|0\rangle$ state and the control atom is not already in the Rydberg state, thus avoiding a Rydberg blockade. The second π pulse then immediately returns the target atom to the ground state.

In the third and final step, a second laser pulse is applied to the control atom. If the control atom was previously in the Rydberg state, it returns to its initial ground state. The only ground state combination that never reaches the Rydberg state is $|11\rangle$, as all pulses are off-resonant and thus have no effect in this case. This distinction is crucial because reaching the Rydberg state introduces a global phase factor of -1, making $|11\rangle$ unique compared to the other states.

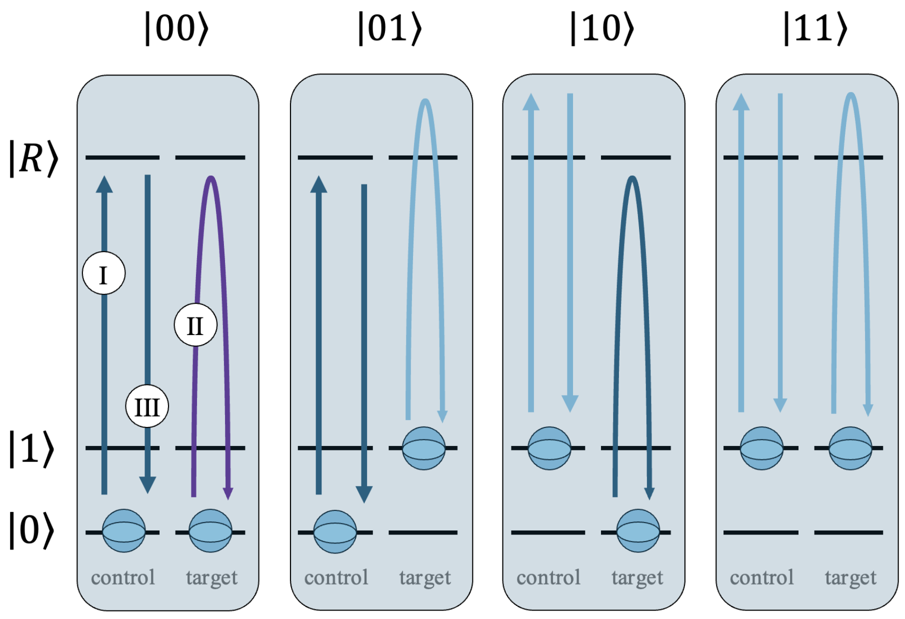


Figure 3.3.: Effect of 3 Laser pulses and the ensuing rydberg blockade on two qubits based on their ground states.Dark blue arrows represent resonant pulses leading to a transition into the Rydberg state, while light blue arrows symbolize off-resonant pulses. The purple arrow at the target qubit in state $|00\rangle$ indicates that, although the pulse would be resonant, it is blocked due to the control qubit being in the Rydberg state.

Building on the CZ gate, other two-qubit gates such as the CX gate can be implemented using additional single-qubit gates.

3.1.3. Register readout

To obtain the results of the computation, a new image of the neutral-atom register must be taken after the process is completed. There are several methods to acquire this information. The first approach is called destructive detection. This method relies on a Charge Coupled Device (CCD) camera, similar to the step used in the 3D configuration of the atom register. Fluorescent light emitted by the illuminated atoms is therefore captured by the camera. However, the key difference in this step is that all qubits in one of the two states are ejected from the tweezers using light of a specific energy. If the $|0\rangle$ state is chosen for ejection, only the qubits that were initially in the $|1\rangle$ state will remain and therefore emit detectable fluorescence. Dark spots thus no longer represent originally empty positions but instead indicate locations where qubits were in the ground state $|0\rangle$. A drawback of the destructive imaging method is that, on average, at least 50% of the tweezers will be empty after the process is complete [24, 25]. Consequently, the entire three-phase process must be repeated for the next computational cycle.

Alternatively, the process of rapid non-destructive readout can be chosen. According to the research by Martinez-Dorantes, "Qubits encoded in the hyperfine ground states of alkali atoms are read out using illumination that resonantly addresses a cycling transition from one ground state while being far detuned from the other ground state" [24]. This means that only qubits in one of the ground states absorb and subsequently emit many photons, making them visible on the imaging device, while qubits in the other state appear as dark spots. This method has a fidelity greater than 98% and ensures that approximately 99% of the atoms remain in the tweezers after the readout process.

3.2. Advantages of the Neutral-Atom Architecture

3.2.1. Long range Interactions and restriction zones

The ability to enable interactions between qubits over long distances is an advantageous feature of certain quantum computer architectures [21]. Superconducting qubits only allow for so-called nearest neighbor connections, meaning interactions can only occur with directly adjacent qubits within the register [14, 26]. On the other end of the spectrum, the trapped-ion architecture provides global connectivity among all qubits in the register. With limited all-to-all connectivity, the NA architecture positions itself between the two extremes. This allows for connections between the central qubit and all other qubits within a specific radius around it. Baker et al's research has identified a maximum interaction radius that is four times the distance between two qubits in the register [14]. However, this interaction radius is determined by the strength of the Rydberg blockade and thus can be improved upon. The strength of the blockade decreases significantly with increasing distance. However, the literature presents varying values for this decrease. Schwarzschild et al. report a scaling of $1/r^3$, while Chi en Wu et al. distinguishes between short distances, where the strength scales as $1/r^3$, and long distances, where it scales as $1/r^6$ [27, 28].

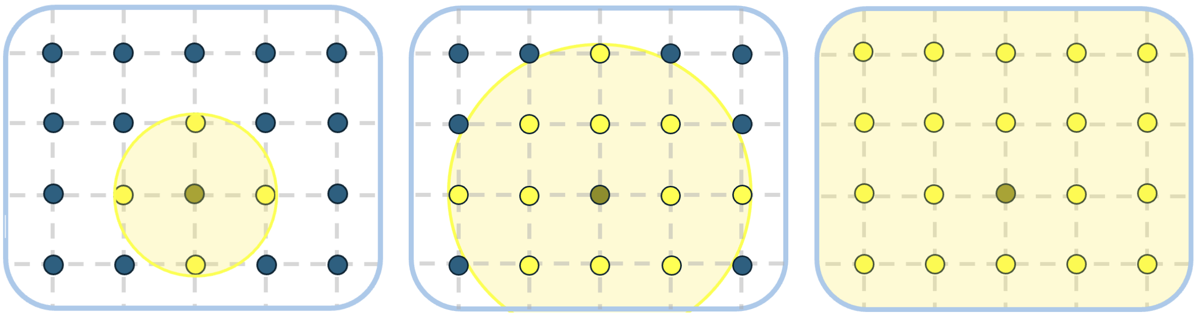


Figure 3.4.: Examples for different Interaction Radii

The primary factors affecting the strength of the Rydberg blockade, and consequently the maximum possible interaction radius, are the chosen Rydberg state and the intensity of the laser [1]. The Rydberg state is uniquely classified by its principal quantum number n. The interaction strength scales with n^{11} [29]. Different Rydberg states thus result in blockades of varying strengths and radii. However, the greater dipole moment, associated with the higher Rydberg states, is also a disadvantage, as it makes them more susceptible to interference [1, 26]. Chi En Wu considers identifying the most suitable Rydberg states essential for achieving large interaction radii [28]. These different states can be achieved through precise manipulation of the Rabi frequency. The formula below relates the interaction strength |C6|, the blockade radius R6, the number of atoms Nb affected by the interaction, and the Rabi frequency [28].

$$\frac{|C_6|}{R_6^b} = \sqrt{N_b}\,\Omega$$

In the case a program requires a long-range interaction between two distant qubits, which is not supported by the hardware architecture, SWAP gates must be introduced. SWAP gates, implemented using three CNOT gates, serve to exchange the states of two qubits. Practically, this allows the state of one qubit to be moved closer to the other qubit, enabling the desired interaction. According to Baker et al., a smaller maximum interaction radius results in any two arbitrary qubits being a larger multiple of the maximum interaction radius apart [14]. To run an algorithm under these conditions, a growing number of SWAP gates must be added as the interaction radius decreases. Henriet et al.'s study illustrates this by examining the number of SWAP gates required for different interaction radii [21]. This number is assessed for randomly generated algorithms, with algorithm length measured by the average number of gates per qubit on the x-axis.

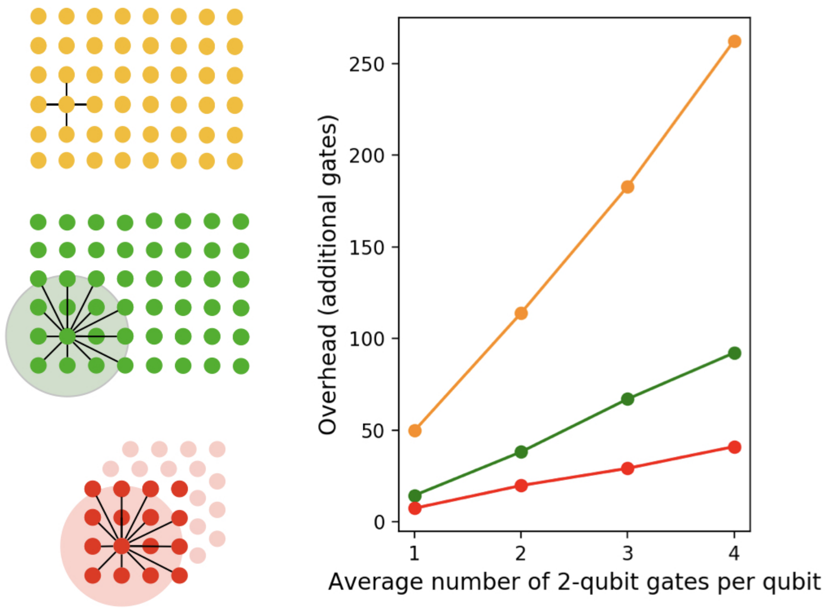


Figure 3.5.: Communication Overhead due to varying maximum interaction distances [21]

The architecture represented in yellow on a 2D register illustrates nearest neighbor connectivity, necessitating by far the largest number of additional SWAP gates. This trend of the highest communication overhead due to SWAP gates is observed regardless of the algorithm's length. Next is the green architecture, which, with an increased maximum interaction radius of 2.3 on the 2D register, requires significantly fewer SWAP gates. Finally, the red architecture, also with a maximum interaction radius of 2.3, is arranged in a 3D cube. This configuration enhances connectivity, as more qubits fall within the interaction range of a randomly selected qubit compared to the yellow or green arrangements. This has the same effect as increasing the maximum interaction radius on a 2D register. Consequently, the red architecture needs even fewer SWAP gates.

The decisive advantage of long-range interactions and the consequent avoidance of SWAP gates lies in a lower program error rate [21]. This error rate can be influenced by two types of errors. The first type is gate errors, which arise from the inherent error rate of each gate during its execution. The second type is known as decoherence errors. These occur over time due to interactions between the qubits and their environment. Programs with longer runtimes are more susceptible to decoherence errors. The time required to execute the entire circuit is described by the circuit depth. Reducing circuit depth, for example by eliminating SWAP gates, can decrease this time. A lower error rate due to less decoherence errors is one benefit, but programs with reduced depth can also be executed more frequently within the same time interval. Due to the statistical nature of measurements, increasing the number of executions can enhance the overall quality of the results. However, the study conducted by Baker et al. reveals that the positive effect of increasing the maximum interaction radius is only noticeable at very small radii [14]. Beyond a radius of 4, no significant improvements in gate count or circuit depth were observed for most algorithms. Nonetheless, the authors note that in much larger circuits, a maximum interaction distance of more than 4 could still lead to significant improvements.

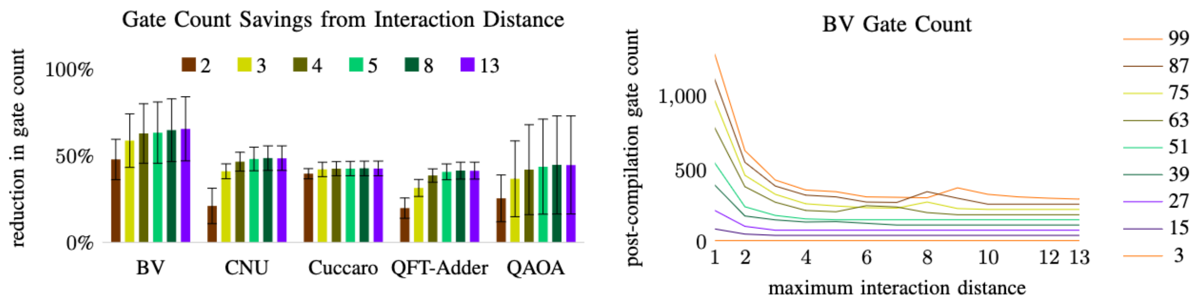


Figure 3.6.: Percentage of gates which could be eliminated through the use of a specific interation radius for a selection of quantum algorithms [14]

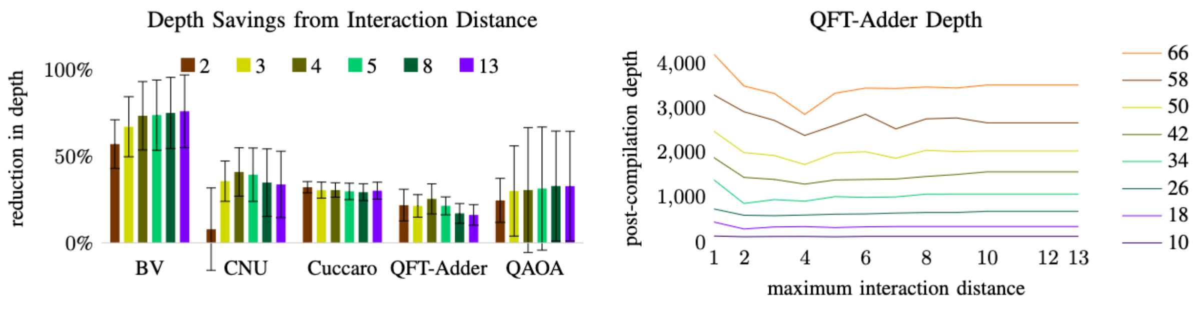


Figure 3.7.: Reduction in depth which could be achieved through the use of a specific interation radius for a selection of quantum algorithms [14]

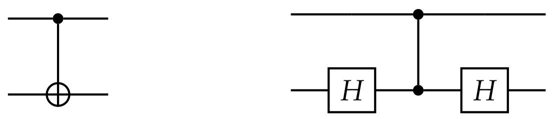
Implementing long-range interactions using Rydberg blockade inevitably introduces the drawback of the restriction zone. This restriction zone means that, aside from the qubits intended to interact via the Rydberg blockade, no other qubits within the zone can perform other interactions or computations simultaneously. Any such interactions would be distorted by the existing blockade, causing unintended interactions with the qubits involved. The radius of the restriction zone r_{re} and the interaction radius r_{int} both depend on the chosen Rydberg state and the strength of its blockade [16]. However, the restriction zone radius is always greater than or equal to the interaction radius. Mathematically, Schmid et al. express this as $r_{\text{re}} = k \cdot r_{\text{int}}$ with $k \geq 1$. A larger restriction zone thus imposes a greater limitation on the underlying atomic register, which restricts the execution of many parallel gates.

According to Baker et al., the limitation of parallelism due to larger restriction zones is mitigated by several factors [14]. Firstly, they note that "many quantum programs are not especially parallel and often do not contain many other gates which need to be executed at the same time" [14]. For instance, Baker et al.'s research identifies significant depth increases due to the enlargement of interaction radii for the QFT-Adder and CNU algorithms. In contrast, the depths of the Cuccaro Adder and Bernstein-Vazirani algorithms are barely affected by an increased interaction radius. Secondly, Baker et al. identify the number of SWAP gates as the dominant cost factor in terms of both gate count and circuit depth. Therefore, the increased depth resulting from a larger interaction radius is often compensated by the elimination of SWAP gates.

3.2.2. Native multiqubit implementation

The capability to natively implement multi-qubit gates incorporating more than two qubits is another significant advantage of the NA architecture. Similar to the two-qubit gates introduced in the background chapter, multi-qubit gates can always be reduced to a combination of control and target qubits. However, the novelty lies in the increased number of qubits, which allows either the transformation on the target qubit to depend on more than one control qubit or the simultaneous transformation of multiple target qubits. Furthermore, given a sufficient number of qubits for the gate, it is possible to design a transformation that integrates multiple control and target qubits at the same time into a single operation. The NA architecture facilitates the direct implementation of various types of multi-qubit gates. One prominent type is the $C_k NOT$ gate, which can have an arbitrary number of control qubits. Isenhower et al. have succeeded in fabricating a gate with k = 35 control-qubits [30]. There are already several physical methods for realizing this type of gate.

The first approach is based on the protocol for CZ gates introduced in the previous chapter. As illustrated in the following diagram, CZ gates can be converted into CNOT gates by placing Hadamard gates before and after the target qubit. To scale the CNOT gate to more than two qubits, it is necessary to find a protocol for executing an arbitrarily large CkZ gate instead of the CZ gate. Adjusting the pulse sequence thereby allows for the integration of more than one control qubit.



The initial strategy for realizing the aforementioned scheme entails defining a sequence in which each control qubit is subjected to a π-pulse one after another. This procedure ensures, that each control qubit has the opportunity, when in the correct ground state, to transition to the Rydberg state and trigger a Rydberg blockade. Subsequently the target Qubit is exposed to a 2π-pulse. The final step of the protocol involves reapplying a π-pulse to each control qubit. Prior to and following the execution of the aforementioned protocol, a Hadamard gate must be implemented on the target qubit through the use of a Raman transition. In summary, the total resource requirements for this protocol are characterized by $2k + 2$ π-pulses and 2 Raman transitions[30].

Isenhower et al. also offer an alternative approach where π-pulses are simultaneously applied to all control qubits. This approach necessitates the selection of a distinct Rydberg state for the control qubits, referred to as $|s\rangle$. This state has the advantageous property of avoiding strong interactions with other qubits in the same state, thus preventing the formation of a Rydberg blockade among qubits also in the $|s\rangle$ state. As a result, multiple control qubits can simultaneously occupy the $|s\rangle$ Rydberg state. Following this, the target qubit is excited via a laser, which transitions it to the Rydberg state $|r\rangle$, depending on its initial state. There is, however, a potential for a Rydberg blockade between the $|r\rangle$ and $|s\rangle$ states, which allows for the conditional transformation inherent to the $C_k Z$ gate to be implemented. In the final step, all control qubits in the $|s\rangle$ state are returned to their original ground state by a single application of the laser tuned to this Rydberg state. By parallelizing parts of the pulse protocol, this method ensures that, regardless of the number of control qubits, the total number of required pulses is limited to 4 π-pulses and 2 Raman transitions for the Hadamard gates.

Another independent aspect that can be adjusted in the physical implementation of this type of gate is the manipulation of the target qubit. Saffman et al. introduce an alternative approach that bypasses the use of two Hadamard-gates and the 2π pulse on the target qubit [31]. Instead, once all π-pulses have been applied to the control qubits, either serially or in parallel, an Amplitude Swap Gate is used on the target qubit. If none of the control qubits has induced a Rydberg blockade, the Amplitude Swap Gate swaps the amplitudes of the target qubit's ground states $|0\rangle$ and $|1\rangle$, effectively performing an X-gate operation. The implementation of this gate requires three π-pulses. Consequently, the total pulse count for this approach amounts to $2k + 3$ π-pulses if the control qubits are excited sequentially. Conversely, when the control qubits are excited simultaneously, the total number of required π-pulses is reduced to just 5.

The following table presents a summary of the various native protocols for $C_k NOT$ gates and their respective costs. The summary is categorized by the method of pulse application to the control qubits and the type of transformation implementation on the target qubit

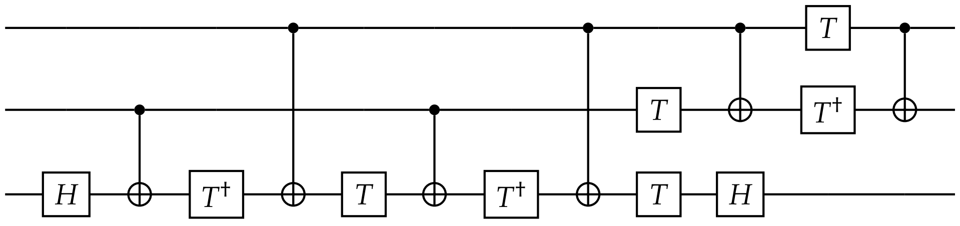
	Realization of Transformation on target Qubit	
	Z-Gate with two Hadamard-Gates	CX via Amplitude swap Gates
Serial Application of **π**-pulses on control-qubits	**2*K π-pulses** **2 π-pulses + 2 Raman Transitions**	**2*K π-pulses** **3 π-pulses**
Simultaneous Application of **π**-pulses	**2 π-pulses** **2 π-pulses + 2 Raman Transitions**	**2 π-pulses** **3 π-pulses**

Figure 3.8.: Pulse costs of various implementation methods for multi-qubit gates on a neutral atom quantum computer

If a quantum algorithm requires a multiqubit gate which isn't supported by the Hardware-architecture, it must be decomposed into a series of single-qubit and two-qubit gates [32]. Maslov and Dueck have demonstrated that for gates with $k \geq 5$ control qubits, the number of required single and two-qubit gates is bounded by the formula $32k - 96$ [33].

Furthermore, Shende and Markov established a lower bound for estimating the resource consumption of Toffoli gates. This bound applies to any number of control qubits k, showing that at least $2k + 2$ CNOT gates are necessary for decomposing any $C_k NOT$ gate [32]. However this lower bound can only be reached if the use of additional ancillae-qubits is permitted. In case the use of ancillae qubits is not authorized the number of required CNOT-gates scales quadratically with the number of involved qubits. These lower bounds highlight that even the most resource-intensive native implementation of multiqubit gates offers significant improvements compared to the most efficient decomposition methods currently available.

This can be clearly seen with the classical Toffoli gate, which is a crucial component in many quantum algorithms [30, 33]. The circuit depicted below represents Shende and Markov's decomposition of the Toffoli gate, optimized for the minimal number of CNOT gates [32]. This decomposition employs 6 CNOT gates, costing 18 π-pulses, and requires 9 Raman transitions for the realization of the single-Qubit Gates. In comparison, the most resource-intensive native implementation of the same gate demands only 7 π-pulses and eliminates the need for Raman transitions.



As the number of control qubits k increases, it can be observed that the factor by which the number of π-pulses can be reduced remains larger than the value of 3, even under the most costly native implementation. This significant reduction in π-pulses has a favorable impact on several key factors essential for the efficient and successful operation of quantum algorithms. According to Schmid et al., the duration for single-qubit gates is 0.5 µs, for CZ gates is 0.2 µs, and for CCZ gates is 1 µs [16]. This implies that a native implementation of the CCX gate would take just 1.4 µs, whereas the decomposed approach depicted above would require 4.2 µs. Moreover, Baker et al. highlight that "efficient decomposition of multiqubit gates often demands a large number of additional ancilla qubits" [14]. By implementing these gates natively, the need for ancillary qubits could be eliminated, leading to more compact quantum circuits.

3.2.3. Scalability of Neutral Atom Quantum Computers

Another critical aspect in assessing the potential usefulness of a quantum computing architecture is its scalability. Alarcón et al. argue that "addressing any real-world problem will require upscaling to thousands or even millions of qubits" [34]. This challenge is seen as one of the fundamental issues in quantum computing, encompassing a broad range of technical hurdles. However, according to Henriet et al., the neutral-atom architecture offers a significant advantage in terms of scalability [21]. This advantage stems from the fact that the size of the quantum register and consequently, the number of available qubits, is directly tied to the number of optical tweezers, each capable of holding a single qubit. Since these tweezers are created using lasers, increasing their number simply requires enhancing the laser's power. As Saffman puts it, "This is a technical and economic challenge, not a fundamental one" [25]. Alternatively, the entire qubit register can be divided into multiple zones, with individual lasers responsible for smaller regions. In this scenario, the required laser power could be limited. Beals et al. conclude that, based on the lasers already available in 2008, it is conceivable to achieve NAQCs with up to 10^6 qubits in a three-dimensional register [35]. This is in stark contrast to other quantum computer architectures, which, according to Graham et al., "require fabrication of completely new chips or traps to increase qubit number" [11].

Beyond laser power, numerous other factors could limit the scalability of the NAQC model. Historically, one significant challenge was the stochastic loading process, which has since been largely addressed [35, 14]. Initially, each optical tweezer had only about a 50% chance of successfully capturing a qubit, making it exponentially harder to fill a larger register without innovative solutions. Nonetheless recent advancements have allowed several teams to demonstrate defect-free assembly of 2D registers with up to 100 qubits [36, 37]. Barredo et al. estimate that with modest improvements in laser power, transport efficiency, initial loading rates, and vacuum-limited lifetimes, their method could scale to allow loading of defect-free structures with up to 1,000 qubits in a 50x50 trap array. Moreover, they report that arrays with up to 1,000x1,000 microlenses have already been produced, potentially allowing for even greater scalability.

Another prerequisite for scalable quantum computers is the ability to perform non-destructive qubit measurements. This capability, known as quantum nondemolition (QND) state measurement, is crucial. Saffman et al. stress the importance of not losing the measured atom and ensuring that the states of neighboring qubits remain unaffected [31]. This ability is essential for quantum error correction procedures [25]. The most widely used method for measuring qubits, described earlier, involves detecting fluorescence photons emitted by qubits in one of the two ground states. Saffman notes that a particular variant of this method initially removes all qubits in one of the ground states, then detects fluorescence from the remaining qubits [25]. However, the drawback of this approach is that it leaves about 50% of the register empty, posing a scalability bottleneck due to the time-consuming reloading process. The non-destructive readout technique introduced by Martinez et al. marks a significant improvement, as it avoids the need to eject half of the atoms. By employing "adequate detuning of the state detection beam" and utilizing "image analysis with Bayesian inference," this method offers a more efficient approach [24].

Despite these advancements, challenges remain for the scalability of quantum computers, particularly in achieving high gate fidelity and minimizing atom loss.

4. Multi Qubit Gate Synthesis

4.1. Goal of multiqubit Gate Synthesis

Most computations on quantum computers are represented using quantum circuits [38]. These circuits consist of individual quantum gates, each of which is uniquely defined by a unitary matrix. Consequently, an entire quantum circuit can be described by a single unitary matrix, derived from the proper composition of the unitary matrices of its constituent gates.

The aim of multi-qubit gate synthesis is to create a new circuit that produces a unitary matrix nearly identical to that of the original circuit. Schmid et al. formalize this process as the search for a gate sequence $\tilde{U} = g_{N-1} \circ \cdots \circ g_0$, where each gate $g_0, \ldots, g_N \in \Sigma_{\text{native}}$ is drawn from the set of native gates Σ_{native} available on the platform [16]. The goal is, given an arbitrary original matrix $U \in \mathbb{C}^{2^n \times 2^n}$, to satisfy the equation $U \approx \tilde{U}$, with only a negligible error.

A minimal difference between the unitary matrices ensures that the synthesized circuit performs nearly the exact same computation as the original circuit. At the same time, the new circuit should incorporate multi-qubit gates to optimize resource usage and realize the various benefits discussed earlier. This concept is illustrated in the images below, where the unitary matrix of the original circuit is shown on the left, and the closely related unitary matrix of the synthesized circuit on the right.

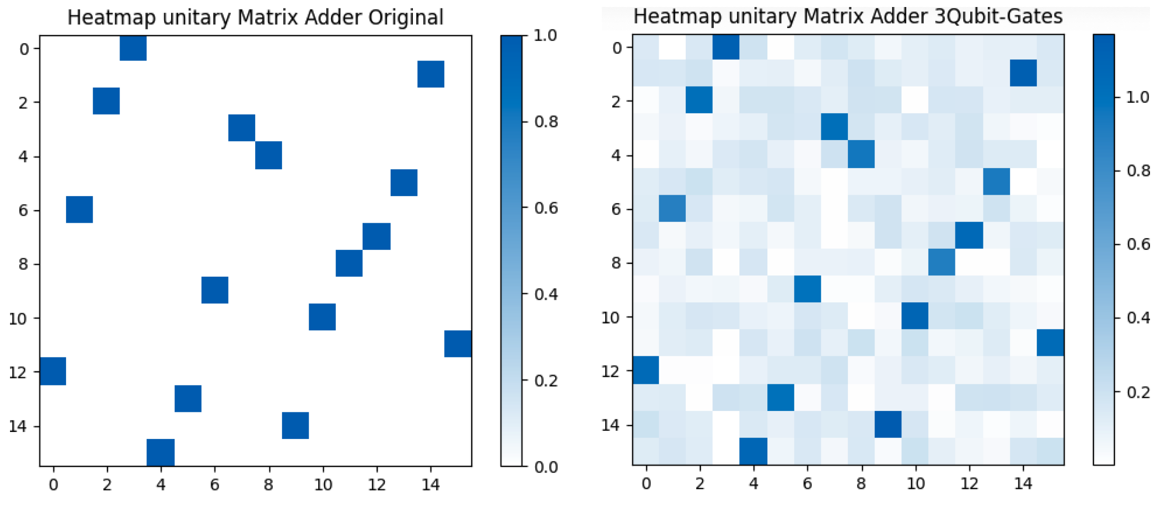


Figure 4.1.: (left) Original unitary Matrix (right) Synthesized unitary Matrix

4.2. Architecture of the Geyser Framework

The framework that serves as the foundation for the experiments conducted in this research is the Geyser Framework [15]. It comprises three key steps: Circuit Mapping, Circuit Blocking, and Block Composition, each of which will be analyzed in detail in the following subsections.

4.2.1. Circuit Mapping Step

According to the authors of the Geyser framework, the mapping process involves translating a logical quantum circuit into a physical circuit [15]. The resulting physical circuit must adhere to the specific constraints and architecture of the NA platform.

The first constraint concerns the permitted gate set, which the authors have limited to the U3 and CZ gates. Consequently, all other logical gates must be decomposed into combinations of U3 and CZ gates within the physical circuit. Another constraint involves the incorporation of SWAP gates. These are necessary to bring qubits that need to interact in the logical circuit—yet are initially placed too far apart in the physical layout—within the interaction range of one another. The third piece of mapping-relevant information pertains to the spatial arrangement of qubits. The Geyser framework exploits the flexibility offered by the NA architecture in spatially organizing qubits within a 2D register. The authors specifically opted to arrange the qubits in a regular triangular topology. The superiority of this approach becomes apparent when comparing it to a quadratic topology, where forming groups of three equidistant qubits is not possible. Executing a multi-qubit gate in such a quadratic layout would require a larger interaction radius, leading to a larger restriction zone. The equidistant arrangement of the triangular topology, on the other hand, allows for increased parallelism and fewer blocked qubits, provided that the multi-qubit gates involve no more than three qubits. As illustrated below, attempting to implement a three-qubit gate in a rectangular topology would block 11 qubits, whereas the triangular topology would block only 9.

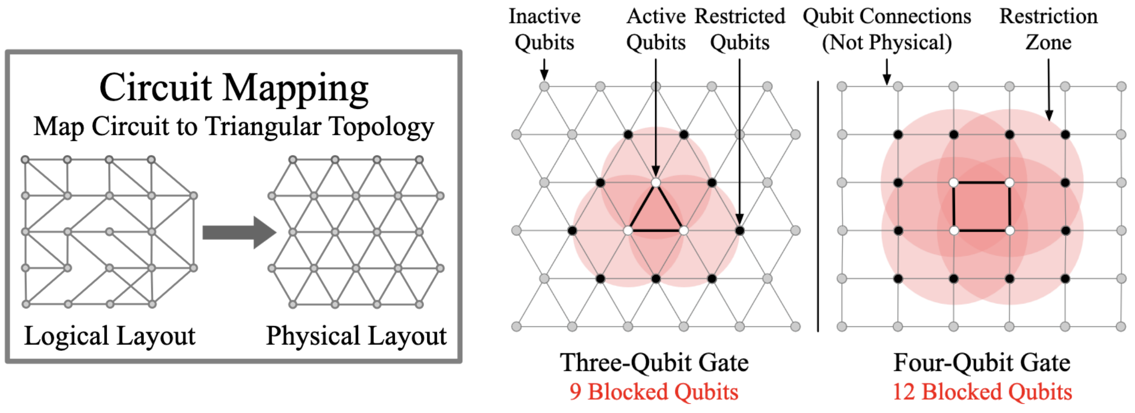


Figure 4.2.: (left) General procedure of the mapping process (right) Mapping process on various regular lattice structures [15]

For the mapping process, the authors leveraged existing techniques provided by the Qiskit compiler [39]. This was feasible because no gates involving more than two qubits had been added to the circuit at this stage. The required compile-parameters included the allowed gate set and the spatial topology of the qubits, along with their connectivity constraints. The Qiskit compiler automatically handled the SWAP gate insertion as part of this process.

4.2.2. Circuit Blocking Step

Circuit blocking involves the deliberate partitioning of the physical circuit, generated in the previous step, into smaller sub-circuits or blocks. In this process, it is not the qubits themselves that are assigned to specific blocks but rather the computational operations. A defining feature of these blocks is that all qubits within a block do not interact with qubits outside of that block throughout its execution. Consequently, if the qubit sets of two blocks overlap, the blocks must be executed in strict sequence. Notably, there are numerous possible ways to implement circuit blocking for any given circuit.

The quality of any partitioning can be assessed based on two key criteria. First, the extent to which it permits parallel execution of blocks. A high degree of parallelism, achieved by minimizing overlaps between the restriction zones of the qubit sets, leads to a shorter overall program duration. Second, the size of the formed blocks, measured by the number of pulses, is crucial. Larger blocks facilitate easier conversion into an equivalent representation using multi-qubit gates in the following synthesis step. However, these two goals—maximizing parallelism and block size—are often at odds with each other. To address this, the authors designed the algorithm to select, at each iteration, the partitioning that contains the maximum number of pulses. An iteration includes all blocks that can be executed immediately, without having to wait for another block due to overlaps. Consequently, both a larger number of blocks and generally longer blocks can become the decisive factor. The pseudo-code for this algorithm, as provided in the Geyser paper, is available in the appendix.

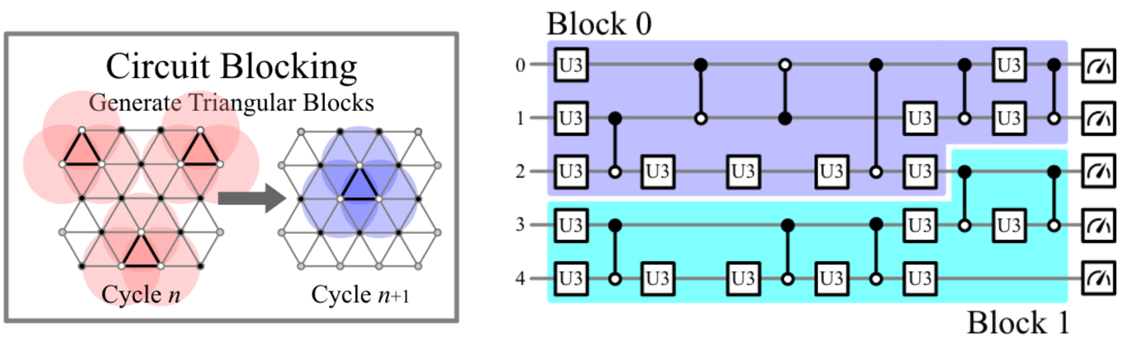


Figure 4.3.: (left) General procedure of the Blocking process (right) Example circuit partitioned into blocks [15]

4.2.3. Block Composition Step

Expansion phase

The composition process follows a cyclical sequence of phases, beginning with an "expansion phase". During this phase, additional gates are incorporated into the circuit's schematic. Initially, a U3 gate is applied to each of the three qubits, followed by a CCZ gate that connects all of them, and then another layer of U3 gates. In subsequent rounds of this phase, only a CCZ gate followed by a layer of U3 gates is added, so that these layers alternate within the composed circuit. This alternation arises because two consecutive U3 gates can always be merged, where the new rotation angles are simply the sums of the corresponding angles from each gate. This structure is also depicted in the visualization below.

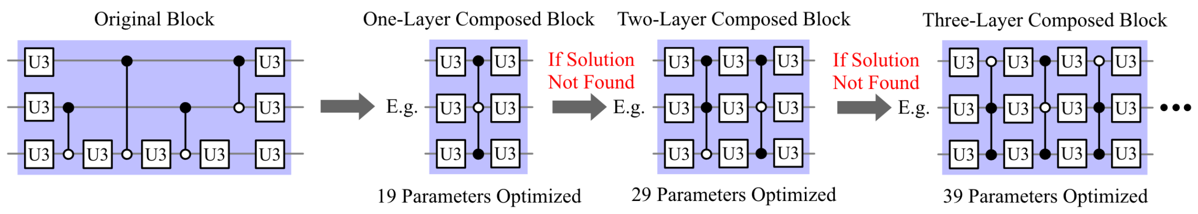


Figure 4.4.: Iterative construction of the synthesized circuit using CCZ and U3 gates [15]

Computation phase: Theory

Next comes the computation phase, where the parameters of the composed block are optimized to ensure that its unitary matrix closely resembles that of the original block. The specific quality metric used here is the Hilbert-Schmidt distance, which quantifies the similarity or distance between two matrices. The Hilbert-Schmidt inner product is calculated as the trace of the product of the complex conjugate transpose of the unitary matrix from the first circuit with the unitary matrix of the second circuit:

$$\mathrm{Tr}(U_1^\dagger U_2)$$

The result of this calculation ranges from 0 to 2^n, where 2^n indicates that U_1 and U_2 are identical, implying no difference. Here, n denotes the number of qubits. To provide a quality measure that is comparable across circuits with different qubit counts, the authors introduced the HS distance. This is done by dividing the HS inner product by the maximum value 2^n and then subtracting it from 1, so that a value of 0 now represents the minimal distance:

$$1 - \frac{\left\|\mathrm{Tr}(U^\dagger U')\right\|}{2^n}$$

The parameters to be optimized in the composition process depend on the number of U3 and CCZ gates. For each U3 gate, three rotation parameters must be specified, each within the range of 0 to 2π. Each CCZ gate, on the other hand, has only one parameter, which is the position of the target qubit. This can be any value between 0 and 2, where 0 indicates that the first qubit is the target. Notably, this parameter must be rounded. This is because a target qubit can't be placed on the 1.67th qubit, so the value must be rounded to the nearest integer, which in this case is 2. In contrast, the rotation parameters determined during the computation phase can be directly copied, as they can be realized with very high precision within the U3 gates.

Computation phase: Dual Annealing

The most critical aspect of the entire framework lies in the fine-tuning of the parameters previously described. This is achieved through an optimization method known as "Dual Annealing", a stochastic approach particularly well-suited for locating minima within high-dimensional landscapes [40, 41]. Stochastic methods have a key advantage over deterministic ones, as highlighted by Xiang et al., in that they are much less likely to get stuck in local minima [42]. Harold Szu illustrates this with the metaphor of a ball rolling across a hilly landscape within a box [43]. The box must be shaken sufficiently for the ball to escape from local minima, yet gently enough so that it eventually settles in a global minimum. The ability to escape local depressions is especially critical when dealing with complex, high-dimensional functions that possess numerous local minima. This makes them especially challenging to navigate using deterministic methods.

The "Dual" in the method's name hints at its two-layered structure. The first layer is a global search algorithm, typically based on the Generalized Simulated Annealing approach. As explained by Xiang et al., this approach is a hybrid, blending Classical Simulated Annealing with Fast Simulated Annealing techniques [42]. However, the true elegance of this method isn't in its specific implementation, but in the conceptual inspiration drawn from the annealing process in materials science. The purpose of annealing in solids is to transition the material to a state of minimal energy, characterized by a highly ordered structure, similar to a crystal lattice [40]. This process begins by heating the material to a high temperature, making it highly malleable and allowing for extensive reconfigurations. The material is then gradually cooled, which reduces the likelihood of random structural changes.

In the optimization context, this translates to the algorithm exploring the search space more aggressively at high temperatures, even if this leads to poorer outcomes. Step 3 of the pseudocode for the Dual Annealing approach clearly illustrates that "annealing allows perturbations to move uphill in a controlled manner" [40]. However, the likelihood of this occurring depends on the temperature and the magnitude of the worsening in the state. Lower temperatures and greater negative changes result in a reduced acceptance probability. As the algorithm "cools", it gradually hones in on the most promising candidates for the global optimum, with fewer and fewer random deviations as it narrows its focus.

Dual Annealing Algorithm Outline [44, 45]

1. **Initialization:**
   - Choose a random starting location x and set $x_{\text{best}} = x$.
   - Select a monotonically decreasing sequence of temperature values $(T_t)_{t \in \mathbb{N}}$, depending on the specific simulated annealing variant [43]:
     - Classical simulated annealing: $\frac{T_a(t)}{T_0} = \frac{1}{\log(1+t)}$
     - Fast simulated annealing: $\frac{T_c(t)}{T_0} = \frac{1}{1+t}$ (faster cooling)
   - Set $t = 0$
2. **Local Variation:** Determine a new point y. The specific method for selecting new points is implementation-dependent, but typically they are chosen in the neighborhood $y \in U(x)$ of the current point.
3. **Selection:**
   - **Case 1:** If the new point has better quality/a lower objective function value $f(y) \leq f(x)$, set $x = y$.
   - **Case 2:** If the new point has worse quality/a higher objective function value $f(y) > f(x)$, set $x = y$, but only with probability $\exp\left(-\frac{f(y)-f(x)}{T_t}\right)$.
4. **Update:** Increment t by 1: $t = t + 1$. Save the current best solution if $f(x) < f(x_{\text{best}})$, and set $x_{\text{best}} = x$.
5. **Local Search:** Decide whether the last point is a good candidate for a global optimum and, if so, initiate a local search.
6. **Decision:** If the stopping condition based on resource consumption is met, terminate the process; otherwise, return to Step 2.

The step of Simulated Annealing can also be visualized in the following graphic, which illustrates a 2-dimensional function and the steps toward the optimum. Red points represent explorations made during the high-temperature phase, while blue points indicate those made during the low-temperature phase.The more extensive exploration observed at high temperatures and the potential for movement against the gradient, as previously discussed, are exemplified here. The latter phenomenon occurs, since some points with lower temperatures are situated above points with higher temperatures.

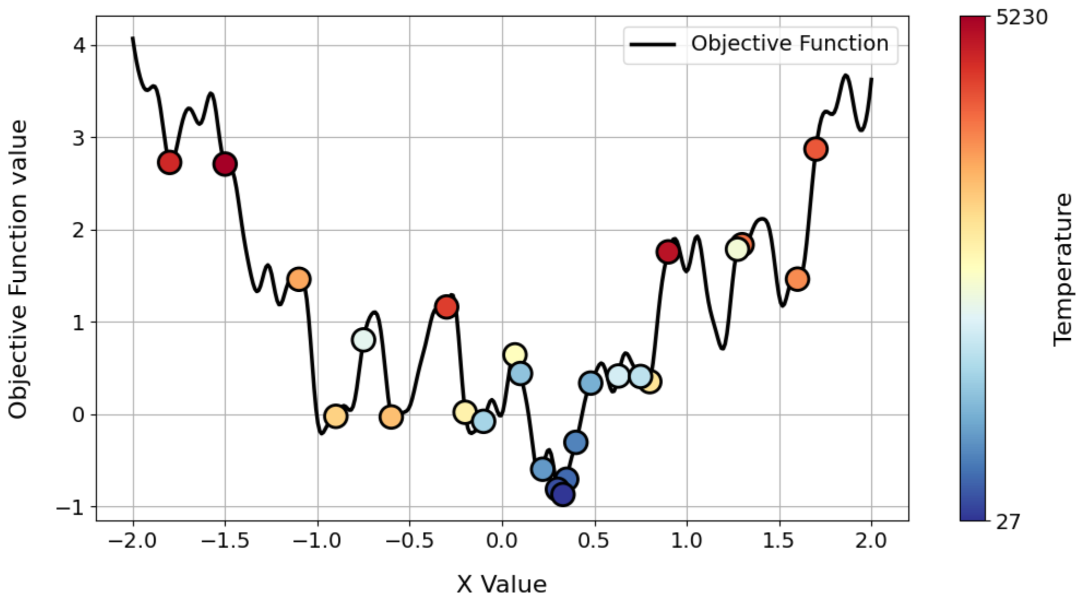


Figure 4.5.: Exemplary Progression of the simulated Annealing Process

Once the global search algorithm identifies a strong candidate for the global optimum, the process transitions to a local search phase. In the Geyser framework, this is handled by the "L-BFGS-B" algorithm.

Decision phase

Following the computation process is the decision phase. First, the total resource consumption in terms of pulse count is determined based on all the gates in the circuit. As mentioned in Chapter 3, each single-qubit gate requires one pulse, each CZ gate requires three pulses, and each CCZ gate requires five pulses. If the total pulse count exceeds that of the original circuit, the process is aborted, and the original block implementation is retained. Otherwise, it is checked whether the Hilbert-Schmidt distance between the unitary matrices is now below the set threshold. If this is the case, the process is successfully concluded, and the composed block replaces the original one. Alternatively, the cycle restarts with the expansion phase, adding another layer to the composed block. The following graphic provides a schematic representation of the entire composition process.

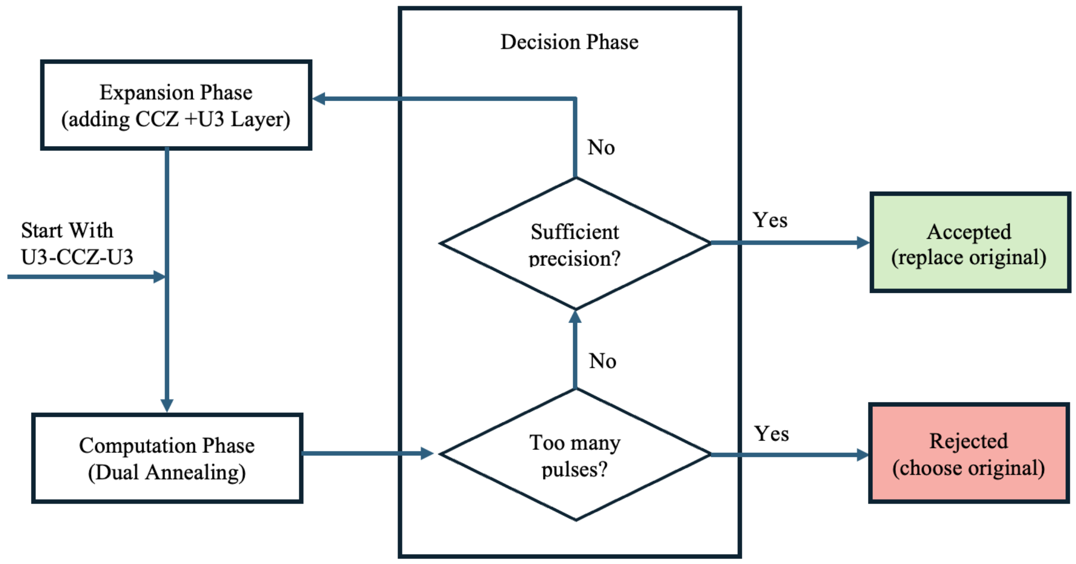

Decision Phase
Expansion Phase (adding CCZ +U3 Layer)
No
Start With U3-CCZ-U3
Sufficient precision?
Yes
Accepted (replace original)
No
Computation Phase (Dual Annealing)
Too many pulses?
Yes
Rejected (choose original)

Figure 4.6.: Simplified Representation of the Decision Procedure from the Geyser Framework

4.3. Architectural Modifications and Scripts for 4-Qubit Gate Synthesis

4.3.1. Architectural Modifications

Due to the modular structure of the Geyser Framework, only minimal adjustments are needed to transition from the existing synthesis of 3-qubit gates to 4-qubit gates. These architectural changes primarily concern the mapping step. The first task involved altering the set of conditions that determine the presence of neighboring qubits in eight cardinal and intercardinal directions. These conditions are calculated for each qubit's position. For instance, a qubit positioned in the top right corner of the 2D register would only satisfy the conditions to its left, bottom-left, and below, since no qubits are present in the other neighboring positions. Additionally, the process for determining qubit IDs had to be revised. This process calculates the IDs of the eight neighboring qubits based on the qubit's own ID. In the previously used triangular pattern, only six neighboring IDs needed to be computed.

With these foundational calculations in place, all connections between qubits were established. These connections represent the physical layout and define the constraints the compiler must adhere to. Leveraging the long-range interactions discussed in Chapter 3, this implementation assumes that qubits can interact with diagonally adjacent qubits, despite the increase in distance by a factor of $\sqrt{2}$. Finally, the block generation process was adjusted. Here, care was taken to always group qubits into blocks of four, ensuring that all theoretically possible block configurations were captured. The previously defined exclusion zones were retained, as blocks of four qubits, as long as they are arranged in a rectangle, do not require larger restriction zones beyond the immediate neighbors. This can be seen on page 21.

4.3.2. Experimental Methodology

The authors of the Geyser Framework indicate that their experiments were conducted using a "local data center consisting of Intel(R) Xeon(R) CPU E5-2690 v3 @ 2.60GHz nodes." Each node was equipped with 128 GiB of RAM and 24 physical or 96 logical cores. Although the exact number of nodes was not explicitly stated, we estimate, based on the results obtained in the subsequent chapter, that the number of nodes utilized was at least 10.

For the experiments conducted in this work, various nodes provided by TUM within the LRZ cluster were employed. The specifications of these nodes are detailed in the Evaluation chapter. A Bash script is indispensable for utilizing the nodes of the LRZ cluster. An example script can be found in the appendix.

Additionally, a list of all modules used can be found in the Appendix. Due to significant changes introduced with the release of Qiskit 1.0.0, following the release of the Geyser Framework, it is advisable to use a virtual environment when running the scripts.

5. Evaluation

5.1. Key Quality Metrics of the Synthesis Process

The Geyser Framework offers a wide range of adjustable parameters, necessitating a careful consideration of which parameters, and subsequently which combinations thereof, should be explored. Additionally, it is essential to evaluate the quality of the results based on specific metrics. We have thus decided to use Maximum Function Calls, Computation Duration, and Pulse Count as the key variable parameters. The combination of the HS-Distance and Total Variation Distance provides a robust means of evaluating the obtained solutions with respect to dependent characteristics. The rationale behind the selection of each individual parameter is presented in the following subsections. For each group of parameters, a dedicated chapter will explore what constitutes an ideal value, along with an examination of the obtained results and the values that are realistically attainable.

5.1.1. Maximum Function Calls, Maximum Iterations & Computation Duration

One of the most crucial criteria for the viability of any program or calculation is its runtime. A division of the Geyser framework into a Mapping & Blocking Phase and a Composition Phase revealed that the latter exhibits runtime variations that are drastically dependent on the chosen parameters. This phenomenon is illustrated in the graph below by the rapidly increasing blue bars. For the maximum number of function calls (maxfun), a value of 1e1 (10) was used for the left bar across all algorithms. The middle bars were calculated with 1e2 (100), while the right bars utilized 1e3 (1000). A consistent maximum of 1000 iteration steps was set, as defined by the default maxiter parameter. Calculations were performed on an M1 Mac equipped with 8 GB of RAM and 8 CPU cores. Although the resulting maximum value of approximately 500 seconds seems relatively low, the Geyser paper standardly uses a value of 1e8 for maxfun. Thus, the extent of computation time and its development are of crucial importance for the feasibility of the algorithm on hardware that does not meet the specifications of a data center.

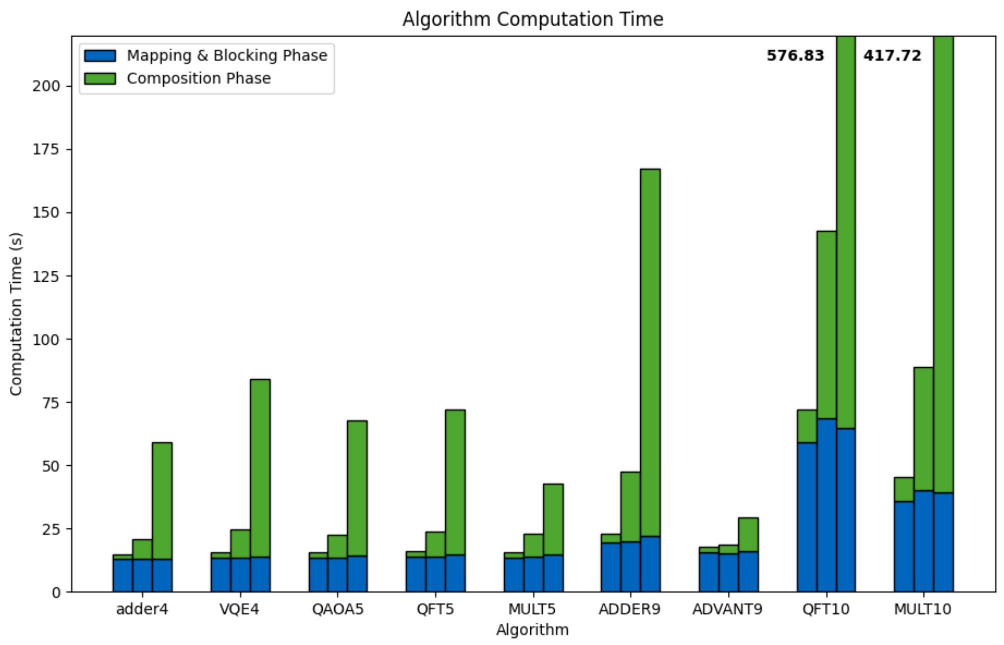


Figure 5.1.: Computation time for various quantum algorithms with 1e1 (left), 1e2 (center), and 1e3 (right) maximum allowed global function calls

As evident from the previous discussion and the accompanying graph, the central factors influencing computation time are the permitted maximum number of function calls and the maximum iterations. To reach this conclusion, it was necessary to examine all parameters of the Python implementation of the Dual Annealing computation method. These parameters are: Iterations, Maxfun, Maxfun_local, Initial_temp, Visit, and restart_temp_ratio. Initial_temp and restart_temp_ratio pertain only to the starting temperature and the amount of cooling required before resetting the temperature to its initial value. Consequently, these parameters have a minimal impact on the total number of computational steps. Their primary effect is on the number of local minima found and, consequently, on the types of computational steps performed. The Visit parameter determines the distribution of points tested by the algorithm (see Chapter 4.2.3, Algorithm Outline, Point 2). In most cases, variations in this value do not lead to a significantly different runtime.

Maxfun_local represents the maximum number of function calls allowed for the local optimization algorithm. It does not affect the total computational resources but restricts how many resources a single local search can consume. With a high value, it is possible that instead of many local searches with limited resources, only a few local searches with extensive resources may be initiated before the resource budget is exhausted. For this reason, the local maxfun parameter can only impact the total computation time if the maximum number of globally allowed function calls significantly exceeds the number of iterations, and the algorithm identifies numerous promising locations for local searches. In such a scenario, a low maxfun local limit would curtail the local search, even though global resources would still be available. Regarding the quality of the solution, the maxlocal parameter can potentially have an impact due to enabling a more extensive local search.

The way in which the Maxiter parameter can influence the overall runtime is analogous to the Maxlocal parameter. Therefore, it must be set appropriately when there is a large number of globally allowed function calls, to avoid artificially constraining the resource budget. For example, if the number of iterations is mistakenly set to 10 and the number of function calls required per iteration due to small blocks is only 25, while 1000 global function calls are available, only 250 function calls would be utilized. During the experiments, such an inappropriate setting of the Maxiter parameter was observed significantly more frequently compared to the Maxfun local parameter. A possible explanation lies in the relatively low frequency with which the Dual Annealing method initiates a local search. This led us to conclude that the Maxiter parameter is one of the two parameters with a significant impact on the overall computation time, warranting further investigation.

Since the global Maxfun parameter directly sets the hard limit for resource consumption, it is clear that it also plays a crucial role in determining both the computation time and the ability to find acceptable solutions for synthesis.

5.1.2. Total Pulse Count

Minimizing the total pulse count is a pivotal objective in the multiqubit gate synthesis process. Achieving the same functionality with fewer pulses not only streamlines execution but also potentially reduces error rates, depending on the fidelity of the multiqubit gate. Yet, more pulses enable the incorporation of additional gates into the composed circuit, which can result in a more accurate representation of the original circuit. To balance these factors, we treat pulse count as a variable parameter, setting it to a specific value to ensure a reduction in the total pulse count. We then explore whether synthesis remains viable with these constraints.

5.1.3. Total Variation Distance & Hilbert Schmidt Distance

As outlined in the Composing chapter, the Hilbert-Schmidt distance is employed to assess the similarity between two unitary matrices. The authors emphasize its advantage, particularly its "lower computational overhead" compared to other metrics [15].

Additionally, this study employs the Total Variation Distance (TVD) as a secondary measure. The TVD is calculated as half the sum of the absolute differences between two probability distributions:

$$\frac{1}{2}\sum_{k=1}^{2n} |p_1(k) - p_2(k)|$$

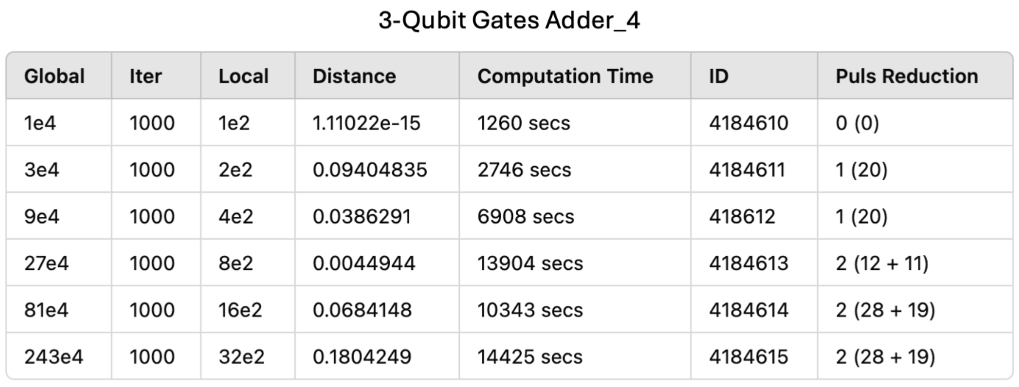
It is crucial to understand that the TVD is applicable only for comparing the probability distributions of all possible outcomes k for a specific input, such as 3, between the original and the composed circuit. Thus, to fully assess the differences between two unitary matrices, the TVD must be computed for every possible input. To enable comparisons across matrices of varying sizes and different sets of possible inputs, the arithmetic mean of these distances can be used.

Together, these two metrics are central to evaluating the quality of the synthesis process and are essential for assessing the effectiveness of circuits synthesized with 4-qubit gates.

5.2. Influence of the Maximum Function Calls & Computation Time

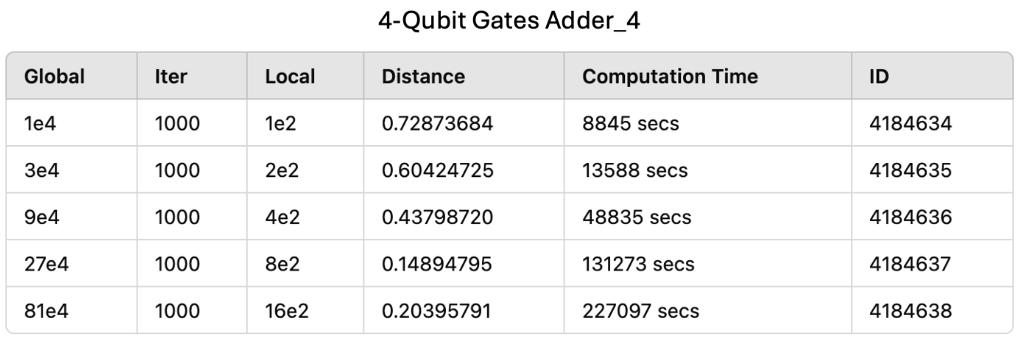
This chapter explores how varying resource allocations affect both computation time and the attainable HS-distance. To this end, six different resource budgets were established. The smallest allows the annealing process a maximum of 10.000 global function calls and 100 local calls. For each subsequent budget, the number of global calls was tripled, and local calls were doubled. The number of maximum iterations remained fixed at the default value of 1.000, as previous studies showed this parameter was never a limiting factor, and the original Geyser framework did not modify it. All experiments were conducted on the "serial_std" partition of the LRZ cluster, utilizing 50 cores, 50 GB of RAM, and a maximum runtime of 96 hours.

The quantum algorithms targeted for optimization included the addition algorithm with 4 and 9 qubits, the variational quantum eigensolver with 4 qubits, and the advantage algorithm with 9 qubits. However, only the results for the 4-qubit algorithms are presented here. The 9-qubit addition algorithm could not be computed in most cases due to excessive time requirements, exceeding the 96-hour limit. Regarding the 9-qubit advantage algorithm, the conclusions of the Geyser framework authors apply even for the 4-qubit synthesis: if the algorithm's structure does not permit long blocks without interactions with external qubits, multi-qubit synthesis is generally ineffective and yields little to no improvements. This was similarly observed in the 4-qubit gate synthesis.



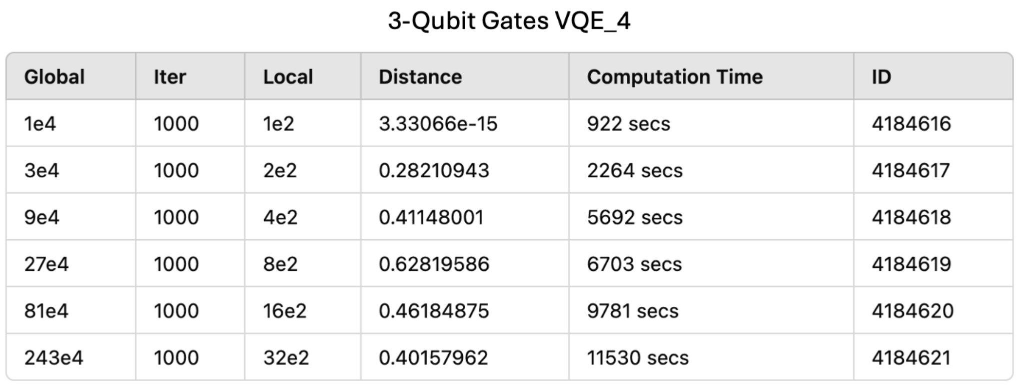
3-Qubit Gates Adder_4

Global	Iter	Local	Distance	Computation Time	ID	Puls Reduction
1e4	1000	1e2	1.11022e-15	1260 secs	4184610	0 (0)
3e4	1000	2e2	0.09404835	2746 secs	4184611	1 (20)
9e4	1000	4e2	0.0386291	6908 secs	418612	1 (20)
27e4	1000	8e2	0.0044944	13904 secs	4184613	2 (12 + 11)
81e4	1000	16e2	0.0684148	10343 secs	4184614	2 (28 + 19)
243e4	1000	32e2	0.1804249	14425 secs	4184615	2 (28 + 19)



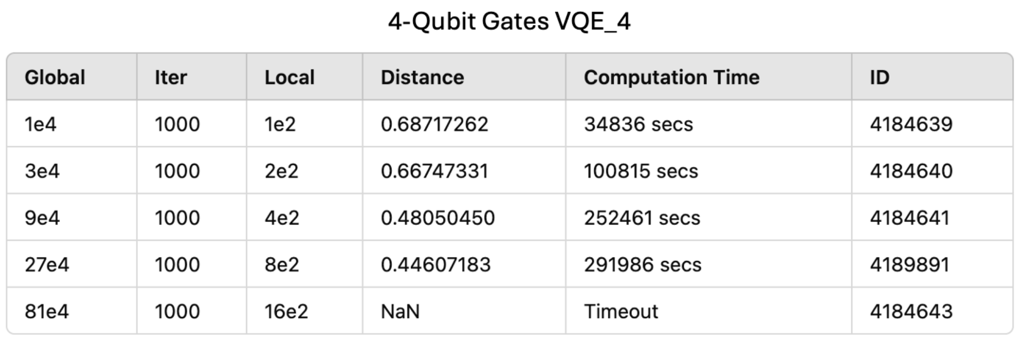
4-Qubit Gates Adder_4

Global	Iter	Local	Distance	Computation Time	ID
1e4	1000	1e2	0.72873684	8845 secs	4184634
3e4	1000	2e2	0.60424725	13588 secs	4184635
9e4	1000	4e2	0.43798720	48835 secs	4184636
27e4	1000	8e2	0.14894795	131273 secs	4184637
81e4	1000	16e2	0.20395791	227097 secs	4184638



3-Qubit Gates VQE_4

Global	Iter	Local	Distance	Computation Time	ID
1e4	1000	1e2	3.33066e-15	922 secs	4184616
3e4	1000	2e2	0.28210943	2264 secs	4184617
9e4	1000	4e2	0.41148001	5692 secs	4184618
27e4	1000	8e2	0.62819586	6703 secs	4184619
81e4	1000	16e2	0.46184875	9781 secs	4184620
243e4	1000	32e2	0.40157962	11530 secs	4184621



4-Qubit Gates VQE_4

Global	Iter	Local	Distance	Computation Time	ID
1e4	1000	1e2	0.68717262	34836 secs	4184639
3e4	1000	2e2	0.66747331	100815 secs	4184640
9e4	1000	4e2	0.48050450	252461 secs	4184641
27e4	1000	8e2	0.44607183	291986 secs	4189891
81e4	1000	16e2	NaN	Timeout	4184643

A recurring trend, also evident in later chapters, is that increasing the resource budget generally leads to improved distance values. However, in the case of 3-qubit gate synthesis, more resources can occasionally produce worse results. This happens when the method, with greater resources, finds an equivalent 3-qubit or 4-qubit gate representation that still contains an error. In earlier experiments with fewer resources, this error was larger, causing the new circuit for that block to be discarded in favor of the original circuit, which had an error of 0 and thus no distance.

This phenomenon can be seen in the 3-qubit gate synthesis for ADDER_4. For example, the run with 243e4 global function calls had almost ten times the calls of the 27e4 run. However, the resulting distance of 0.1804 was significantly higher than the 0.0045 achieved in the lower-resource run. This discrepancy is explained by the fact that the resource-constrained attempt only eliminated 23 pulses, whereas the higher-resource trial managed to optimize 47 pulses. Nevertheless, a fundamental observation across all test series is that a larger resource budget generally correlates with better results.

A notable trend is that the distances achieved with 4-qubit gate synthesis frequently resemble those obtained with 3-qubit gate synthesis. This is particularly evident in trials 3 and 4 of the VQE_4 algorithm, where the distances are 0.628 (3-qubit) compared to 0.480 (4-qubit) and 0.461 (3-qubit) compared to 0.446 (4-qubit), respectively. For the ADDER_4, a more noticeable disparity is observed in the fifth trial, with distances of 0.068 (3-qubit) and 0.204 (4-qubit). Nevertheless, these distances remain comparable according to the previously mentioned explanation.

A key consideration for the feasibility of 4-qubit gate synthesis is its runtime. This runtime increases as more resources are allocated. Specifically, in 4-qubit gate synthesis, the rise in runtime correlates closely with the factor by which the limit on global function calls is increased. The fourth trial of the VQE_4 algorithm with 4-qubit gate synthesis is the only instance with an unusually small increase in computation time. If this trial is disregarded, the average growth rate of runtime for the two algorithms using 4-qubit gate synthesis is 2.47. The increase in computation time for 3-qubit gate synthesis often does not align directly with the expansion factor of the resource budget. This might be due to the fact that processing multiple blocks can better utilize hardware and tap into unused capacity. On average, the factor by which computation time grows is around 1.635.

More critical than this growth rate is the comparison between 4-qubit and 3-qubit gate synthesis with the same resource budget. For instance, in the fifth trial of both methods, the 4-qubit gate synthesis for the ADDER_4 algorithm requires 227.097 seconds, which is nearly 22 times longer than the 10.343 seconds needed for the 3-qubit synthesis. Similarly, for the VQE_4 algorithm, the time factor is even more pronounced, nearing 44. These substantial differences in time consumption and less favorable growth factors suggest that achieving an HS-distance of 0.01 or lower while using the 4-qubit gate synthesis demands a disproportionately large increase in resources for all algorithms.

5.3. Influence of the Total Pulse Count

The experiments focusing on the Total Pulse Count aim to investigate how the achievable distance changes when the synthesis algorithm is permitted to utilize more pulses, thereby increasing the number of multi-qubit gates. In this context, a total of five trials were conducted on the "teramem_inter" partition of the LRZ cluster. The setup specified 96 cores, 4000 GB of RAM, and a maximum runtime of 96 hours. While the first trial was limited by a maximum of 1.25×10^4 allowed function calls, the limit was doubled for each subsequent trial. As a result, the fifth experiment had access to a total of 2e5 permitted function calls. All other parameters, such as the local Maxfun value or the number of iterations, remained fixed at 5e2 and 1000, respectively.

The algorithm that was approximated in this setup using 4-qubit gate synthesis is the Variational Quantum Eigensolver (VQE) with 4 qubits. In its original configuration, the algorithm utilizes 457 pulses. However, since the original Geyser framework already achieves a reduction to approximately two-thirds of the original pulse count, this study only examined the progression up to 300 pulses. The results from the five experimental series are presented in the figure below and will be analyzed in greater detail in the following paragraph.

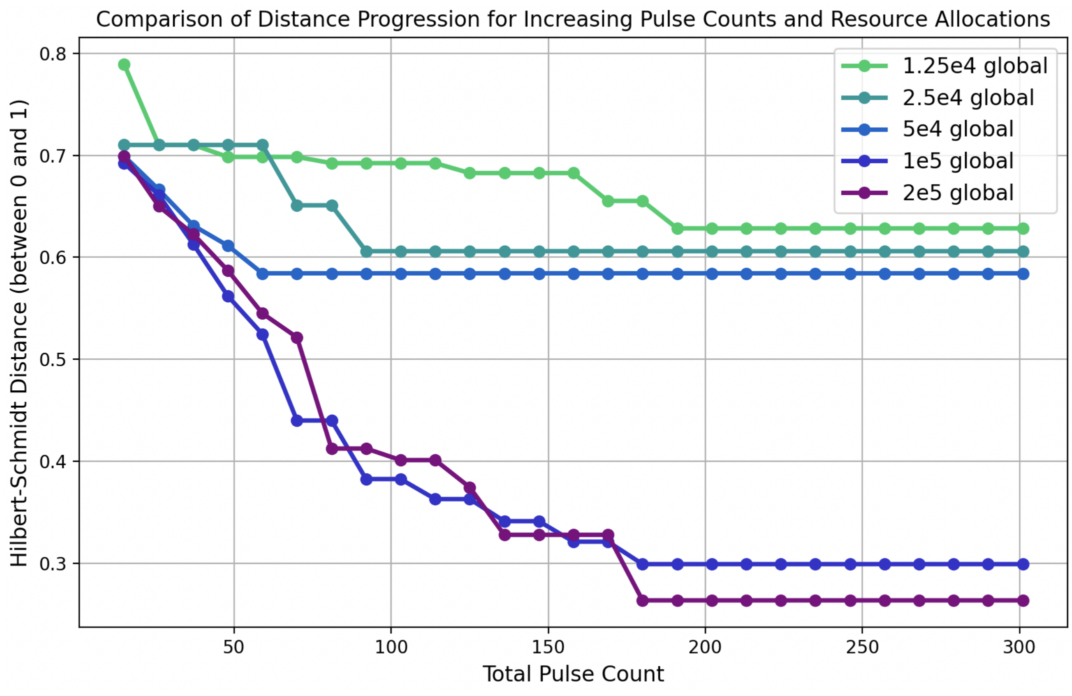


Figure 5.2.: Comparison of the evolution of the HS-distance with gradual increases in the amount of allowed pulses for different computation time budgets

For all experiments, it is evident that the HS distance can be minimized at some point by increasing the budget of allowed pulses. This is expected, as with more pulses the annealing process gains more flexibility to adjust the composed unitary matrix towards the original unitary matrix. However, the changes in pulse quantities, the extent of improvement, and the achievable minimal distance vary among the experiments.

A general trend is that most improvements occur when extending the pulse sequence while the total pulse count is still below half of the maximum allowed. Beyond 200 pulses (two-thirds of the maximum), no further optimization was observed in any trial. The phase where the distance stagnates and cannot be further optimized will be referred to as the "plateau phase" in the following discussion. The experiments differ significantly in when they reach this plateau phase. For the trials with 2.5e4 and 5e4 pulses, this happens relatively early, at 59 and 92 pulses, respectively. In contrast, trials with either higher or lower resource budgets enter the plateau phase much later.

One possible explanation is that, beyond a certain number of pulses, the function call budget is insufficient to explore the entire parameter space. More pulses thus hinder rather than facilitate finding an optimal solution, as the algorithm becomes "overwhelmed" by the increased number of combinations. Visually, one can imagine that for each pulse count and corresponding gates, an optimal configuration exists that minimizes the distance between unitary matrices. With more pulses, the potential to minimize the distance further exists, but significantly more resources are needed to find this optimal configuration. Once the number of pulses requires excessive resources to find the optimal configuration, the algorithm is only able to reproduce the previously found optimal solution. Finding a suitable alternative circuit with multi-qubit gates and a very low number of pulses thus seems unlikely, as especially the three experiments with the most ressources show more or less similar values for for the VQE_4 with fewer than 75 pulses. Increasing the resource budget in this range appears to be rather ineffective.

Except for the trial with the lowest resource budget, all other experiments achieve an HS distance within 0.02 of 0.7 in the first iteration. Despite similar starting points, improvements vary significantly. Notably, trials with larger resource budgets consistently show better quality measures upon reaching the plateau phase. The most substantial improvement is seen between the third and fourth trials, with the latter reducing the minimum achievable distance from 0.58 to 0.299. Other adjacent trials generally differ by less than 0.03. Thus, a larger resource budget appears essential for finding viable alternatives with 4-qubit gates. However, based on the results, it is not possible to precisely predict whether a usable HS distance of 0.001 or less can be achieved or the amount of resources required.

Before reaching the minimal distances, it is observed that experiments with lower resource budgets sometimes achieve smaller distances than those with higher budgets. For instance, the series with 1e5 pulses for 114 pulses has a value of 0.362, which is better than the series with 2e5 pulses at 0.401. This discrepancy can be attributed to the stochastic nature of the dual annealing approach, which cannot guarantee the attainment of a specific threshold with certainty.

5.4. Implications of the achievable Total Variation Distance & Hilbert Schmidt Distance

The viability of the synthesis method using 4-qubit gates depends on how closely the results match the original circuits. A universal bound for HS-distance or TV-distance applicable to all algorithms cannot be established. Instead, each algorithm must be individually assessed to define its specific acceptance criteria. Equally relevant is the observation from the previous chapter that HS distance reflects an average of deviations. This arises from calculating output deviations for every possible input and then averaging them. For instance, an HS distance of 0.4 might result from each input having a deviation of 0.4, or from some inputs exhibiting high deviations of 0.7, balanced by other inputs with deviations as low as 0.1. Given the way HS distance is calculated, a value of 0.1 does not always correspond directly to a deviation rate of 0.1. Only in the case of the later discussed "deterministic" algorithms these metrics align. Therefore, this metric should be seen as a rough estimate. A more meaningful conclusion can be drawn from the TVD, provided it is calculated individually for each possible input. This directly reveals the proportion of outcomes that do not align with the desired distribution.

The following evaluates the quality of distance values using the addition algorithm as an example. As depicted in Figure 4.1, which shows the algorithm's unitary matrix, it is a "deterministic" algorithm. This means that a given input always maps to the same output. The evaluation of these distance values can thus be extended to other deterministic algorithms, such as the quantum multiplication algorithm discussed in the Geyser paper. The benefit of deterministic algorithms is that the error rate is calculated as the sum of probabilities leading to incorrect outputs, rather than just shifts in certain probabilities. Incorrect and correct results are thus in disjoint sets. If the output is correct with a probability of $50\% + \epsilon$ and the individual tests are stochastically independent, the error rate can be further reduced through repeated testing combined with a majority vote. For example, with $\epsilon = 0.05$ (yielding a correctness probability of 0.55), $n = 518$ trials are needed to achieve a 99% confidence level in the majority result. For $\epsilon = 0.15$ and $\epsilon = 0.25$, only 52 and 16 trials, respectively, are required. This calculation can be verified using the formula below. Alternatively, the values can be approximated using the Chernoff bound.

Let X be a binomially distributed random variable representing the number of correct results. We seek the value of n such that

$$P(X > 0.5n) \geq 0.99.$$

This probability can be expressed as:

$$P(X > 0.5n) = \sum_{i=\lceil 0.5n \rceil}^{n} \binom{n}{i} \cdot p_{\text{correct}}^{i} \cdot p_{\text{incorrect}}^{n-i},$$

where p_{correct} is the probability of a correct result and $p_{\text{incorrect}}$ is the probability of an incorrect result.

We therefore need to find n such that:

$$\sum_{i=\lceil 0.5n \rceil}^{n} \binom{n}{i} \cdot p_{\text{correct}}^{i} \cdot p_{\text{incorrect}}^{n-i} < 1 - \text{significance level.}$$

A drawback of the majority vote is the need for frequent execution of the circuit. This negates the benefits that were intended to be gained through the use of 4-qubit gates.

An HS distance of 0.15 could be considered acceptable for a deterministic algorithm, provided the TVD for each individual input closely matches this 0.15. The critical factor is the input with the highest TVD. As previously mentioned, inferring TVD from HS distance is only applicable to "deterministic" algorithms due to their calculation method. Additionally, there must be a willingness to perform $n = 6$ repetitions per trial and to consider an error rate of 0.01 as acceptable.

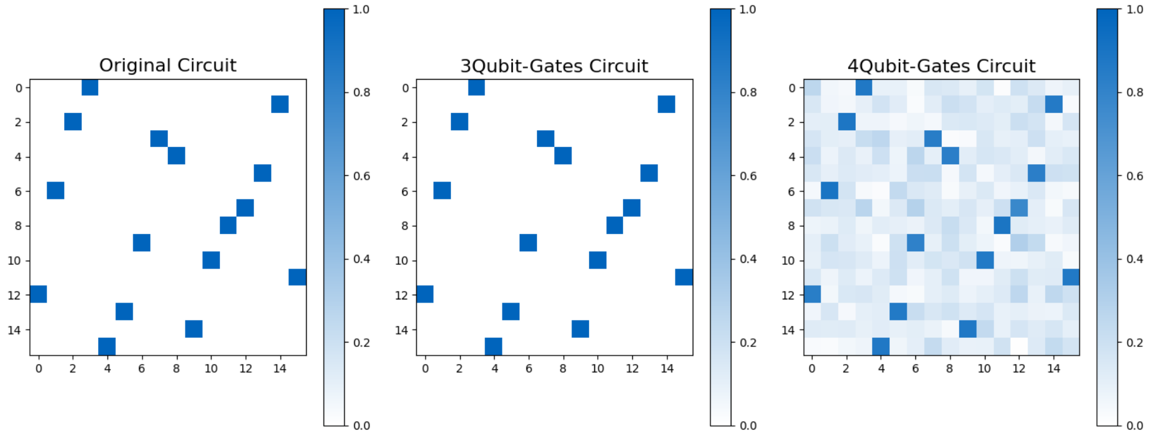


Figure 5.3.: Comparison of the original unitary matrix with the matrices obtained through the two synthesis methods

The best HS distance achieved for the Adder_4 algorithm, using 2.7e5 maxfun global, 8e2 maxfun local and 1000 Iterations over 36.5 hours, was 0.149. To contextualize this best error rate for 4 Qubit Gate synthesis, we compare it with the corresponding 3-qubit gate circuits from the original Geyser framework. For this comparison, the original settings of 1e8 maxfun global, 1e4 maxfun local and 1000 Maxiter were used. The resulting HS distance of 1.226e-10 highlights a significant difference.

A thorough examination analyzed the TVD for each of the 16 possible inputs from 0000 to 1111, with each input tested 10.000.000 times. The results showed that no incorrect outputs were observed in any of the trials. Therefore, the 3-qubit gate circuit, obtained with relatively few resources, matches the original precisely, with only a negligible deviation. These results indicate that the 4-qubit gate circuits achievable with current computational resources do not meet the standards of the 3-qubit circuits. This discrepancy is illustrated in the previous image, which compares the unitary matrices of the original circuit, the 3-qubit circuit, and the 4-qubit circuit. The significant differences between the 4-qubit gate circuit and the 3-qubit circuit are clearly visible. However, many patterns of similarity are also apparent. For example, each row consistently features the correct single entry with the highest probability, a characteristic inherent to deterministic algorithms, which the 4-qubit gate synthesis has managed to capture. Thus, there is potential that with a substantial increase in resources or further optimizations in computation time, the results of the 4-qubit circuit could approach those of the original Geyser framework.

6. Conclusion & Outlook

6.1. Conclusion

Neutral atom quantum computers represent a novel architecture within the realm of quantum computing, offering several advantages that have drawn significant attention from the scientific community. Among these are their capability to facilitate long-range interactions and the native implementation of multi-qubit gates. However, a significant drawback is the loss rate of atoms serving as qubits, particularly during gate operations. This creates a need for algorithms that minimize the number of gates and pulses used. The Geyser Framework addresses this by attempting to reduce gate and pulse counts through the use of 3-qubit gates, specifically employing U3 and CCZ gate combinations. The goal is to design circuits that approximate the original with minimal error.

This work focused on whether the Geyser Framework could be extended to accommodate 4-qubit gates. To this end, modifications to the mapping procedure within the Geyser Framework were made. Additionally, preparations were completed to utilize the high-performance computing resources of the TUM-LRZ cluster.

A series of computationally intensive tests followed. The first experiment evaluated the theoretically achievable HS distances, along with the corresponding computation time, for a given resource allocation. Results showed that the HS distances obtainable with the same amount of resources were comparable between the original 3-qubit synthesis approach and the newly introduced 4-qubit synthesis. However, the 4-qubit approach required disproportionally more time, with factors ranging from 20 to 50 times greater. The conclusion of this experiment is that further optimization will necessitate significantly increased resource investment.

The second experiment series examined the number of pulses required to achieve optimal distance values. It was observed that when resource budgets were insufficient, an increased number of available pulses and gates could not improve the distance. For higher resource budgets, it became evident that reducing pulses by more than 50% is unlikely to succeed.

In the final section, a sample calculation was presented to estimate which HS distances might be practical for the subset of "deterministic" quantum algorithms. The upper limit for a majority decision was identified as a maximum TVD of 0.5 for each input. However, practical applications would require circuits with distances as low as 0.01, a value not yet achieved with the 4 qubit synthesis process. For comparison, the original Geyser Framework demonstrated distances as low as 10^{-8}.

6.2. Outlook

In summary, without further optimizations in resource consumption or distance, the current results of the 4-qubit gate synthesis for the algorithms tested in this study are deemed impractical. However, a final assessment of the viability of 4-qubit gate synthesis will inevitably require the examination of larger quantum algorithms. Additionally, the use of alternative approximation algorithms could be of interest. The method currently employed for local search in the dual annealing process could potentially be replaced by more effective alternatives.

A. General Addenda

The following link leads to a GitHub repository containing all files relevant to this work. This includes the result text files from both experimental series, as well as the Geyser framework in both its original form and my modified version. Additionally, the repository contains a file detailing my adjustments and bug fixes in the Geyser code. I have also included a list of all used packages and the Python files for generating my plots.

https://github.com/Philippwon/Bachelor-Thesis-Computer-Science-Multiqubit-Gate-Synthesis.git

List of Figures

Bibliography

[1] 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 (2023), p. 32.

[2] R. P. Feynman. “Simulating physics with computers”. In: *Feynman and computation*. CRC Press, 2018, pp. 133–153.

[3] J. Preskill. “Quantum computing 40 years later”. In: *Feynman Lectures on Computation*. CRC Press, 2023, pp. 193–244.

[4] P. W. Shor. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”. In: *SIAM review* 41.2 (1999), pp. 303–332.

[5] L. K. Grover. “Quantum mechanics helps in searching for a needle in a haystack”. In: *Physical review letters* 79.2 (1997), p. 325.

[6] M. Calixto. “Physical Basis of Quantum Computation and Cryptography”. In: *International Work-Conference on the Interplay Between Natural and Artificial Computation*. Springer. 2007, pp. 21–30.

[7] J. D. Hidary and J. D. Hidary. *Quantum computing: an applied approach*. Vol. 1. Springer, 2019.

[8] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver. “Superconducting qubits: Current state of play”. In: *Annual Review of Condensed Matter Physics* 11.1 (2020), pp. 369–395.

[9] J. A. Jones and M. Mosca. “Implementation of a quantum algorithm on a nuclear magnetic resonance quantum computer”. In: *The Journal of chemical physics* 109.5 (1998), pp. 1648–1653.

[10] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage. “Trapped-ion quantum computing: Progress and challenges”. In: *Applied Physics Reviews* 6.2 (2019).

[11] T. Graham, Y. Song, J. Scott, C. Poole, L. Phuttitarn, K. Jooya, P. Eichler, X. Jiang, A. Marra, B. Grinkemeyer, et al. “Multi-qubit entanglement and algorithms on a neutral-atom quantum computer”. In: *Nature* 604.7906 (2022), pp. 457–462.

[12] D. S. Weiss and M. Saffman. “Quantum computing with neutral atoms”. In: *Physics Today* 70.7 (2017), pp. 44–50.

[13] J. Wurtz, A. Bylinskii, B. Braverman, J. Amato-Grill, S. H. Cantu, F. Huber, A. Lukin, F. Liu, P. Weinberg, J. Long, et al. “Aquila: QuEra’s 256-qubit neutral-atom quantum computer”. In: *arXiv preprint arXiv:2306.11727* (2023).

[14] J. M. Baker, A. Litteken, C. Duckering, H. Hoffmann, H. Bernien, and F. T. Chong. "Exploiting long-distance interactions and tolerating atom loss in neutral atom quantum architectures". In: *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*. IEEE. 2021, pp. 818–831.

[15] T. Patel, D. Silver, and D. Tiwari. "Geyser: a compilation framework for quantum computing with neutral atoms". In: *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 2022, pp. 383–395.

[16] 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 (2024), p. 033001.

[17] K. H. Sahin and A. R. Ciric. "A dual temperature simulated annealing approach for solving bilevel programming problems". In: *Computers & chemical engineering* 23.1 (1998), pp. 11–25.

[18] J. Liu, M. Bowman, P. Gokhale, S. Dangwal, J. Larson, F. T. Chong, and P. D. Hovland. "QContext: Context-aware decomposition for quantum gates". In: *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2023, pp. 1–5.

[19] B. A. Jumayev. "Modelling and testing multi-qubit universal control gate developed for quantum computing systems". In: *Proceedings of the 23rd International Conference on Computer Systems and Technologies*. 2022, pp. 20–27.

[20] Z. Wen, Y. Liu, S. Tan, J. Chen, M. Zhu, D. Han, J. Yin, M. Xu, and W. Chen. "Quantivine: A visualization approach for large-scale quantum circuit representation and analysis". In: *IEEE Transactions on Visualization and Computer Graphics* (2023).

[21] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak. "Quantum computing with neutral atoms". In: *Quantum* 4 (2020), p. 327.

[22] M. Kajita and M. Abe. "Frequency uncertainty estimation for the 40CaH+ vibrational transition frequencies observed by Raman excitation". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 45.18 (2012), p. 185401.

[23] D. P. DiVincenzo. "Two-bit gates are universal for quantum computation". In: *Physical Review A* 51.2 (1995), p. 1015.

[24] M. Martinez-Dorantes, W. Alt, J. Gallego, S. Ghosh, L. Ratschbacher, Y. Völzke, and D. Meschede. "Fast nondestructive parallel readout of neutral atom registers in optical potentials". In: *Physical review letters* 119.18 (2017), p. 180503.

[25] M. Saffman. "Quantum computing with atomic qubits and Rydberg interactions: progress and challenges". In: *Journal of Physics B: Atomic, Molecular and Optical Physics* 49.20 (2016), p. 202001.

[26] 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.

[27] B. Schwarzschild. “Experiments show blockading interaction of Rydberg atoms over long distances”. In: *Physics Today* 62.2 (2009), pp. 15–18.

[28] C.-E. Wu, T. Kirova, M. Auzins, and Y.-H. Chen. “Rydberg-Rydberg interaction strengths and dipole blockade radii in the presence of Förster resonances”. In: *Optics Express* 31.22 (2023), pp. 37094–37104.

[29] E. Urban, T. A. Johnson, T. Henage, L. Isenhower, D. Yavuz, T. Walker, and M. Saffman. “Observation of Rydberg blockade between two atoms”. In: *Nature Physics* 5.2 (2009), pp. 110–114.

[30] L. Isenhower, M. Saffman, and K. Mølmer. “Multibit C k NOT quantum gates via Rydberg blockade”. In: *Quantum Information Processing* 10 (2011), pp. 755–770.

[31] M. Saffman, T. G. Walker, and K. Mølmer. “Quantum information with Rydberg atoms”. In: *Reviews of modern physics* 82.3 (2010), pp. 2313–2363.

[32] V. V. Shende and I. L. Markov. “On the CNOT-cost of TOFFOLI gates”. In: *arXiv preprint arXiv:0803.2316* (2008).

[33] D. Maslov and G. W. Dueck. “Improved quantum cost for n-bit Toffoli gates”. In: *Electronics Letters* 39.25 (2003), pp. 1790–1791.

[34] E. Alarcón, S. Abadal, F. Sebastiano, M. Babaie, E. Charbon, P. H. Bolıvar, M. Palesi, E. Blokhina, D. Leipold, B. Staszewski, et al. “Scalable multi-chip quantum architectures enabled by cryogenic hybrid wireless/quantum-coherent network-in-package”. In: *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE. 2023, pp. 1–5.

[35] T. R. Beals, J. Vala, and K. B. Whaley. “Scalability of quantum computation with addressable optical lattices”. In: *Physical Review A—Atomic, Molecular, and Optical Physics* 77.5 (2008), p. 052309.

[36] D. Ohl de Mello, D. Schäffner, J. Werkmann, T. Preuschoff, L. Kohfahl, M. Schlosser, and G. Birkl. “Defect-free assembly of 2D clusters of more than 100 single-atom quantum systems”. In: *Physical review letters* 122.20 (2019), p. 203601.

[37] D. Barredo, S. De Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys. “An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays”. In: *Science* 354.6315 (2016), pp. 1021–1023.

[38] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Vol. 2. Cambridge university press Cambridge, 2001.

[39] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen, J. M. Chow, A. D. Córcoles-Gonzales, A. J. Cross, A. Cross, J. Cruz-Benito, C. Culver, S. D. L. P. González, E. D. L. Torre, D. Ding, E. Dumitrescu, I. Duran, P. Eendebak, M. Everitt, I. F. Sertage, A. Frisch, A. Fuhrer, J. Gambetta, B. G. Gago, J. Gomez-Mosquera, D. Greenberg, I. Hamamura, V. Havlicek, J. Hellmers, Ł. Herok, H. Horii, S. Hu, T. Imamichi, T. Itoko, A. Javadi-Abhari, N. Kanazawa, A. Karazeev, K. Krsulich, P. Liu, Y. Luh, Y. Maeng, M. Marques, F. J. Martín-Fernández, D. T. McClure, D. McKay, S. Meesala, A. Mezzacapo,

N. Moll, D. M. Rodríguez, G. Nannicini, P. Nation, P. Ollitrault, L. J. O'Riordan, H. Paik, J. Pérez, A. Phan, M. Pistoia, V. Prutyanov, M. Reuter, J. Rice, A. R. Davila, R. H. P. Rudy, M. Ryu, N. Sathaye, C. Schnabel, E. Schoute, K. Setia, Y. Shi, A. Silva, Y. Siraichi, S. Sivarajah, J. A. Smolin, M. Soeken, H. Takahashi, I. Tavernelli, C. Taylor, P. Taylour, K. Trabing, M. Treinish, W. Turner, D. Vogt-Lee, C. Vuillot, J. A. Wildstrom, J. Wilson, E. Winston, C. Wood, S. Wood, S. Wörner, I. Y. Akhalwaya, and C. Zoufal. *Qiskit: An Open-source Framework for Quantum Computing*. `https://doi.org/10.5281/zenodo.2562111`. Accessed: 2024-08-19. 2019.

[40] R. A. Rutenbar. "Simulated annealing algorithms: An overview". In: *IEEE Circuits and Devices magazine* 5.1 (1989), pp. 19–26.

[41] S. Gubian. *Dual Annealing for global optimization problems*. Youtube. URL: `https://www.youtube.com/watch?v=0P4Z4_Csn1s&t=1s`.

[42] Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng. "Generalized simulated annealing for global optimization: the GenSA package." In: *R J.* 5.1 (2013), p. 13.

[43] H. Szu. "Fast simulated annealing". In: *AIP conference proceedings*. Vol. 151. 1. American Institute of Physics. 1986, pp. 420–425.

[44] Wikipedia contributors. *Simulated Annealing — Wikipedia, The Free Encyclopedia*. [Online; accessed 25-September-2024]. 2023. URL: `https://de.wikipedia.org/wiki/Simulated_Annealing`.

[45] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts. *Simulated annealing*. Springer, 1987.