Compiling OpenQASM on a Neutral Atom Quantum Computer

Thesis Proposal

Author: Xavier Spronken (i6225376)

x.spronken@student.maastrichtuniversity.nl

Supervisor: Claire Blackman

Total Word Count: 2000
Text:1794, Captions:178, Mathematical expressions: 28

Academic Year 2022-23

Contents

1 Introduction

1.1 Quantum Computers

While quantum computers can be used for analog computation or simulating quantum systems[4], it is also possible to make them perform quantum information processing (QIP). The carriers of information in QIP are qubits. These differ from classical qubits in that they make use of quantum superposition. Thus a qubit's state (either $|1\rangle$ or $|0\rangle$) can only be known precisely upon measurement. Before this, the qubit is said to be in a superposition of both states, where the probability of measuring either one of those states is $\mathrm{P}(|1\rangle) + \mathrm{P}(|0\rangle) = 1$[15]. Mathematicaly a qubit is represented as $|a\rangle = \alpha|1\rangle + \beta|0\rangle = \begin{bmatrix}\alpha\\ \beta\end{bmatrix}$, where $|\alpha|^2 + |\beta|^2 = P(|1\rangle) + P(|0\rangle) = 1$

A method to view the state of a qubit before measuring is the Bloch sphere. In this representation, the $|0\rangle$ state and $|1\rangle$ state are set to the north and south poles of the sphere respectively[14]. In addition to the probability of getting either state upon measurement, the Bloch sphere shows the phase of the qubit thanks to the latitudinal direction of the arrow in the sphere[6].

Before qubit measurement, one can change the state of a qubit using quantum logic gates, these are the quantum computing equivalents of logic gates that make up classical electronic circuits. When applied to a qubit, a quantum gate will set it to a specific superposition of states. This corresponds to a specific rotation of the arrow on the Bloch sphere[7]. Since a gate corresponds to a rotation it does not place the qubit in a specific state, the resulting superposition depends both on the gate and the state of the qubit before the gate was applied. Some well-known quantum gates are the Pauli x y and z gates, each corresponding to a rotation of 180 degrees around the x, y and z axes of the Bloch sphere. Gates are mathematically represented as matrices that can be applied to the qubit state vector. The Pauli x gate for example would be $\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$.

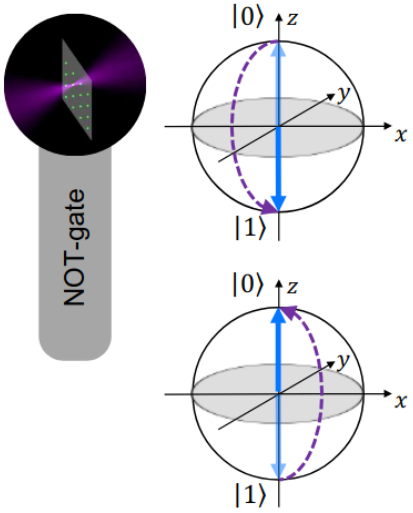

Figure 1: Bloch sphere representation of the effect of a not (or Pauli x) gate[4]

Quantum gates are applied to single or multiple qubits at the same time, and in general, these are control gates, which will only affect the target qubit if the control qubit is already in state $|1\rangle$. While there are multiple different types of gates, it is possible to create a universal set. Gates in this set can be combined to create any other gate. The set of arbitrary rotation gates $R_x(\theta), R_y(\theta), R_z(\theta)$, combined with a phase shift gate $P(\varphi)$ and a control-X gate are such a set[13].

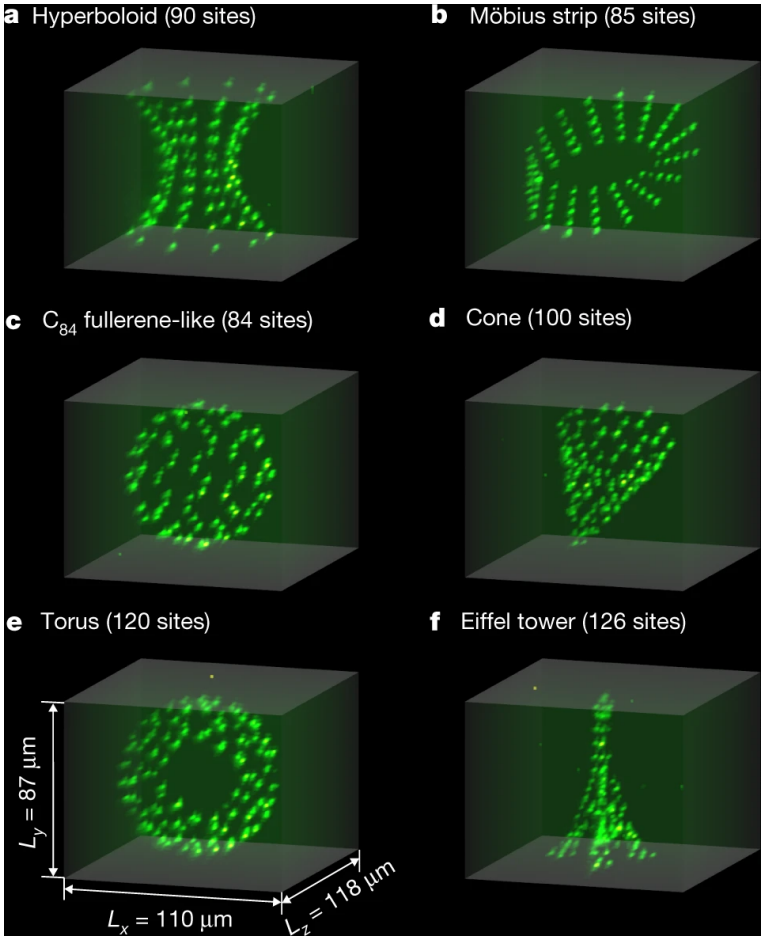

Figure 2: Different qubit configurations in a 3D neutral atom register [1]

Finally, these qubits are stored in a register. Depending on the type of quantum computer the registry can be fixed, such as for superconducting qubits, where all the qubits are pre-engraved onto a chip and used as needed [5]. Other quantum computers such as the Neutral atom quantum computer built by Pasqal, have a dynamic registry, where a chosen number of qubits can be placed, at the user's will, into a 2D or 3D grid. This extends the ability of the device to be able to perform simulations of quantum systems and analog computations that cannot be represented in a traditional quantum circuit.

1.2 QASM

OpenQASM (Open Quantum Assembly Language) is a programming language for quantum circuits that allows the user to create a register of qubits and comes with a preselection of gates, as well as the universal set of arbitrary rotation, phase shift and CX gate. OpenQASM has become the standard for programming quantum circuits, in part due to it being machine-independent, meaning it can be compiled to any quantum computer [3]. Along with the preset gates, custom user-created gates can be defined as well saved to the circuit before compilation and execution. OpenQASM 3 also incorporates timing statements as well as classical computing interactions within a circuit, allowing the user to precisely define when gates or measurements need to be applied, but also allowing the quantum circuit to perform classical operations and interact with a regular processor during the execution of the circuit [3].

1.2.1 stdgates.inc

- file containing the standard gate definitions used to write quantum algorithms
- QASM Unitary gate $\mathrm{Rz}(\lambda)\mathrm{Rz}(\theta)\mathrm{Rz}(\phi)$
- gphase

- gate modifiers (ctrl (important), negctrl,pow,inv (less important))

1.3 Pasqal's Neutral Atom Quantum Computer

1.3.1 Register

This device uses optical beams to create multiple $1\,\mu m^3$ sized traps[10]. Due to their size, these traps can only contain a single atom. The optical beams are then pointed at a vacuum chamber containing atomic vapor and used to single out and reposition atoms into a grid.

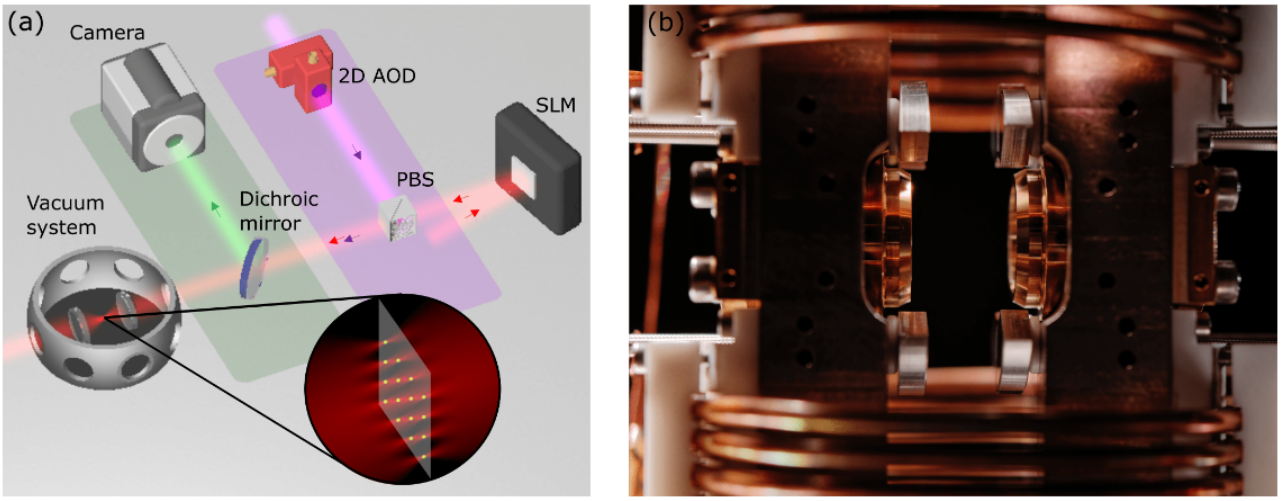

Figure 3: "(a) Overview of the main hardware components constituting a quantum processor. The trapping laser light (in red) is shaped by the spatial light modulator (SLM) to produce multiple microtraps at the focal plane of the lens (see inset). The moving tweezers (in purple), dedicated to rearranging the atoms in the register, are controlled by a 2D acousto-optic laser beam deflector (AOD) and superimposed on the main trapping beam with a polarizing beam-splitter (PBS). The fluorescence light (in green) emitted by the atoms is split from the trapping laser light by a dichroic mirror and collected onto a camera. (b) Photography of the heart of a neutral-atom quantum co-processor. The register is prepared at the center of this setup" [4]

1.3.2 Gates

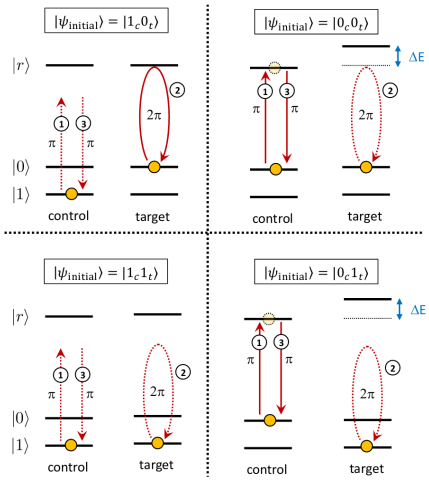

Figure 4: Sequence of optical pulses needed for a Control Z gate[11]

Gates are applied using optical pulses from two different channels: the Raman and Rydberg channels, which can target a single qubit (local) or all of them (global). These pulses are used to attain two different excitation states from the ground state $|g\rangle \equiv |0\rangle$. The Raman pulses are used in single qubit gates to access the hyperfine state $|h\rangle \equiv |1\rangle$[12]. Multi-qubit gates on the other hand require the use of quantum coherence, which is achieved using a phenomenon called the Rydberg blockade: If an atom is excited to the Rydberg state $|r\rangle$[2](using the Rydberg channel of the quantum device), any nearby atom situated within a certain radius will be blocked from achieving the Rydberg state[8]. This interaction is used, for example in the implementation of a CZ gate[9].

1.3.3 Pulser

Pasqal has published a Python library named Pulser that simulates their quantum device. It can create a registry with any kind of pattern as well as the required optical pulses. The pulses have a configurable waveform with amplitude Ω, detuning δ phase shifting φ and the duration τ of the pulse. Manipulating these parameters, it is possible to recreate any single-gate rotation on the Bloch sphere with angles :

$$(\Omega\tau\cos\phi, \Omega\tau\sin\phi, \delta\tau)$$

[4]

1.3.4 Pulser's Unitary gate

- corresponds to $\mathrm{Rz}(\lambda)\mathrm{Rx}(\theta)\mathrm{Rz}(\phi) = \mathrm{Rz}(\lambda+\phi)\mathrm{Rz}(-\phi)\mathrm{Rx}(\theta)\mathrm{Rz}(\phi)$
- In pulser θ is amplitude of pulse, ϕ is phase, λ is post phase shift (explain post phase shift) post phase shift allows for a virtual z-gate to be applied(X and Y rotations are possible natively but not Z)

2 Methodology

2.1 Translating Qubit Gates

Used for all gates: - Uses pulser.Pulse class, with constant detuning equal to 0 , as it is not needed to perform the unitary gate.
- Waveform = pulser.waveforms.BlackmanWaveform, waveform doesn't seem to affect gates, although no rigourous testing was performed.
- Waveforms are defined by "area under curve" and pulse duration, no amplitude needed, and regardless of duration "Area under curve" $= \theta$ as mentioned in introduction. - all same duration @ 250ns. Thus pulse duration will depend on the characteristics of the physical quantum computer.

2.1.1 translating Single Qubit Gates

Pulser has native unitary gate U(λ,θ,ϕ)= $\mathrm{Rz}(\lambda)\mathrm{Rx}(\theta)\mathrm{Rz}(\phi)$= $\mathrm{Rz}(\lambda+\phi)$U($-\phi$,$\theta$,$\phi$)

Two possibilities:
1. translate native Native Unitary Pulser gate = $\mathrm{Rz}(\lambda)\mathrm{Rx}(\theta)\mathrm{Rz}(\phi)$ to QASM = $\mathrm{Rz}(\lambda)\mathrm{Ry}(\theta)\mathrm{Rz}(\phi)$
- Convert the Pulser Rx rotation into Ry rotation by adding $\pi/2$ to the 1st Rz rotations add removing $\pi/2$ from the second rotation.

2. Create a universal set of gates using the Pulser unitary gate and try and reproduce QASM unitary gate through universal set combinations.
Approach:
Create the arbitrary rotation set, which consists of Rx, Ry, Rz, phase gate and the 2-qubit CNOT gate.
$\mathrm{Rx}(\theta)$ = Pulser_U(0,θ,0) or Pulser_U(-pi,θ,pi) if theta negative
$\mathrm{Ry}(\theta)$ = Pulser_U($\pi/2, \theta, -\pi/2$) or ...

Ry(θ) = Pulser_U(θ, 0,0) or ...

phase gate does not seem to have a native implementation to Pasqal's device.

2.1.2 Translating two Qubit Gates

- CZ is only native 2-qubit gate. (raman pi pulse+rydberg 2pi pulse + raman pi pulse) - Can make CNOT by doing H CZ H on qubit

2.1.3 Testing gates

- Using pytest define random initial state and gate parameters.
- Create theoretical gate using matrix form
- Compare Pulser Gate output to Theory Gate output
- Compare the two outputs up to 10^{-4} accuracy, since gate fidelity rarely achieves 10^{-5} - repeat n times
- valid if 100% passrate

2.1.4 Testing Rydberg Blockade

- PLace 3 qubits in a equilateral triangle register with triangle side smaller than rydberg blockade radius
- repeat the gate testing but with a sequence of 3 CZ gates (in Pulser and in theory).
- apply gates consecutively to q0 and q1 then q1 and q2 then q2 and q0
- make sure theoretical output matches pulser output
- if results are correct, I can apply a multi-qubit gate to 2 qubits even though a 3rd qubit is within rydberg radius (only if gates are applied consecutively and not in parallel)

2.1.5 Parsing QASM code

Might skip this since the work I have is not even close to fully functional. - Using AST visitor classes to retrieve information. - Implement syntax error detection in each visitor - Store correct data in a lsit of consecutive tuples - read data from tuples and convert into regsiter creation methods and gate methods.

3 Results

- show annex with screenshots of tests - Successful for all arbitrary rotation gates using native Pulser U gate - successful for 2 qubit gates CZ and CX - unseccessful translation from Pulser_U to QASM_U: $|1\rangle$ state is in the wrong phase. (could be solved if the Phase gate could be implemented, since this gate produces a phase shift around the $|1\rangle$ state). - Rydberg Blockade testing (hopefully successful, will know for certain on 24th may)
- Parsing mostly unsuccessful due to length of the task (show some screenshots of what it can do so far)

4 Discussion

- need to find a way to implement phase gate, if this can be done a universal set is possible, and the QASM_U gate would also be functional
- no need for an excessively complex registry creation algorithm since all qubits can simply be placed within eachothers rydberg radius.
- Machine specs limiting, can only place a limited amount of qubits that close to eachother
- Constraints for optimization would not require to exclude certain qubits from the rydberg radius to operate properly (simplifies mechanism greatly).
- Since only native multi-qubit gate is CZ, an algorithm would need to be developped in order to recreate all user defined gates, based combinations with the CZ gate.
- This makes the ctrl (and negctrl) modifier very hard to implement (since "ctrl @ gate" would have to be somehow iplemented using a combination of CZ and single qubit gates)
- Pow modifier no known way to implementation - inv is easy for single gates (simply reverse rotation on Rx Ry and Rz, which is possible thanks the fact the Pulser implementations accept negative angle paramters)
- Parsing is a fairly straightforward task once AST is understood, however the parser needds to be able to handle every single type of information present in QASM code, which is much rore than initially expected, on top of this user syntax error correction also needs to be implemented wich makes the job even longer.

5 Conclusion

All single gates successful, except Phase (and QASM_U by extension)

easy: registry creation, most single qubit gates hard: making arbitrary multiqubit gates with ctrl modifier, unknown: making phase gate.

Conclusion: While very good for analog computation, the lack of phase gate and native multi-qubit gates make it very hard to implement "digital" computation on this machine, it is possible though. But will require complex multi-qubit gate creation algorithms that are capable of finding a combination of gates that satisfies any user-defined multi-qubit gate. As well as an algorithm that is capable of creating an adequate registry once all the pairs of qubits that interact due to a native CZ gate are found.

6 Critical Reflection

- Project taking much longer than planned and more complex than expected
- Parsing very long, redundant and repetitive work. (need to account for all user errors and syntax etc.) - First time working with very limited documentation and having to discover things through experimenting and analyzing source code of simulation libraries
- initial method of wanting a fully functional pipeline from the get go (even for a simple algorithm) was naive.

References

[1] Daniel Barredo, Vincent Lienhard, Sylvain de Léséleuc, Thierry Lahaye, and Antoine Browaeys. Synthetic three-dimensional atomic structures assembled atom by atom. *Nature*, 561(7721):79–82, September 2018.

[2] Sam R. Cohen and Jeff D. Thompson. Quantum Computing with Circular Rydberg Atoms. *PRX Quantum*, 2(3):030322, August 2021.

[3] Andrew Cross, Ali Javadi-Abhari, Thomas Alexander, Niel De Beaudrap, Lev S. Bishop, Steven Heidel, Colm A. Ryan, Prasahnt Sivarajah, John Smolin, Jay M. Gambetta, and Blake R. Johnson. OpenQASM 3: A Broader and Deeper Quantum Assembly Language. *ACM Transactions on Quantum Computing*, 3(3):12:1–12:50, September 2022.

[4] Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020.

[5] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien. Quantum computers. *Nature*, 464(7285):45–53, March 2010.

[6] David C. McKay, Christopher J. Wood, Sarah Sheldon, Jerry M. Chow, and Jay M. Gambetta. Efficient Z gates for quantum computing. *Physical Review A*, 96(2):022330, August 2017.

[7] Bharadwaj Chowdary Mummaneni, Jing Liu, Georgios Lefkidis, and Wolfgang Hübner. Laser-Controlled Implementation of Controlled-NOT, Hadamard, SWAP, and Pauli Gates as Well as Generation of Bell States in a 3d–4f Molecular Magnet. *The Journal of Physical Chemistry Letters*, 13(11):2479–2485, March 2022.

[8] M Saffman. Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 49(20):202001, October 2016.

[9] M. Saffman, T. G. Walker, and K. Mølmer. Quantum information with Rydberg atoms. *Reviews of Modern Physics*, 82(3):2313–2363, August 2010.

[10] Nicolas Schlosser, Georges Reymond, Igor Protsenko, and Philippe Grangier. Sub-poissonian loading of single atoms in a microscopic dipole trap. *Nature*, 411(6841):1024–1027, June 2001.

[11] Henrique Silvério, Sebastián Grijalva, Constantin Dalyac, Lucas Leclerc, Peter J. Karalekas, Nathan Shammah, Mourad Beji, Louis-Paul Henry, and Loïc Henriet. Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays. *Quantum*, 6:629, January 2022.

[12] Richard Bing-Shiun Tsai, Henrique Silvério, and Loic Henriet. Pulse-level Scheduling of Quantum Circuits for Neutral-Atom Devices, June 2022.

[13] Colin P. Williams. Quantum Gates. In Colin P. Williams, editor, *Explorations in Quantum Computing*, Texts in Computer Science, pages 51–122. Springer, London, 2011.

[14] H. M. Wiseman and G. J. Milburn. Interpretation of quantum jump and diffusion processes illustrated on the Bloch sphere. *Physical Review A*, 47(3):1652–1666, March 1993.

[15] Thomas G. Wong. *Introduction to Classical and Quantum Computing.* Rooted Grove, Omaha, Nebraska, 2022.