Architecture for a microwave trapped ion quantum computer

Victory Omole
Iowa State University
104 Marston Hall 533 Morrill Rd.
Ames, IA 50011-2151
vtomole@iastate.edu

ABSTRACT

Quantum computing is a revolutionary field that has changed the way we think about the theory of computation. Theoretically, these devices can solve certain problems that would be intractable with the classical computers we use today. Although physical implementations of quantum computers have proved to be extremely difficult, researchers have been making slow but noticeable progress in the last two decades. Over the years, there have been plans to build large scale quantum computers that would be able to scale from tens to thousands, and even millions of qubits. There have also been a countless number of architecture designs that would combine these physical implementations with implementation-dependent instruction set architectures. One such design for a large scale quantum computer is the microwave ion trap quantum computer. An instruction set architecture (ISA) for this design does not exist. In this paper I will propose an ISA for this type of quantum computer. One that combines the physical implementation of a trapped ion microwave quantum computer with an ISA designed by Rigetti Computing. This architecture will contain two compilers. The first will transform the implementation-independent ISA to an implementation dependent one. The second will be an error correcting compiler that will transform an implementation-independent ISA to an implementation dependent error correcting ISA.

CCS Concepts

•Computer systems organizations → Quantum computing;

Keywords

Computer architecture, Quantum computing, Compiler

1. INTRODUCTION

Quantum computers can solve certain problems; such as factoring and simulation of quantum systems, which cannot be solved efficiently on classical computers. Quantum computers can be generalized into two categories. Near-term quantum computers, and long term-quantum computers. Near-term quantum computers are computers that will be realized in the next 5 to 10 years. They are composed of tens to hundreds of qubits. Long term quantum computers could potentially be realized in a few decades. They will contain thousands to millions of qubits. This paper will address both types of computers.

Researchers at the Department of Physics and Astronomy at the University of Sussex have recently published a blueprint for a large scale trapped ion quantum computer [5]. This computer can serve as a near-term and long-term quantum computer because it is modular. The only factor that depends on whether or not it is a near-term or long-term computer, is how many individual modules can be added to the hardware before the quality of the qubits worsen to the point where the computer cannot execute its instructions correctly.

This paper will first cover the basics of quantum computing. It will then transition into discussing previous research that was performed in this area. This will be followed by a couple of paragraphs about a new type of architecture for a microwave ion trapped quantum computer. After talking about the benefits of the design, the last two sections will contain a concluding paragraph and a discussion about some potential directions for future research.

2. PRELIMINARIES

2.1 Quantum Mechanics

2.1.1 *Two-level Quantum System*

The fundamentals of quantum theory can be illustrated by the double slit experiment [3]. This experiment is composed of an electron source, a wall with two slits, and a backdrop. When an electron is shot at the wall with slits, there is an interference pattern on the backdrop. This result suggests that electrons are waves. There is a dilemma. The electron left the source as a particle, so which slit did it go through? To answer this question, we place a detector in front of the slits. This detector detects that the electron goes through one slit or the other, not both. The electron's behavior seems to depend on whether or not its state is observed. When it's observed, it behaves like a particle, when not, it behaves like wave. The act of measuring the electron collapses the wave function. The double-slit experiment highlights the wave-particle duality of light. This behavior describes all quantum systems, such as photons and atoms. It cannot be explained using classical physics.

The consequences of the double-slit experiment can also be described as a two-level quantum system. The double-slit experiment and the two-level quantum systems highlight that quantum particles can be in a superposition of states. For example, an atom is composed of two states. A ground state and an excited state. When the atom is not measured, it is in a superposition of the ground state and the excited state. As soon as it is measured, there is a 50 percent chance that the atom will be in a ground state and a 50 percent chance that it will be in an excited state. Quantum mechanics can be described by probability theory.

2.1.2 *Quantum Entanglement*

Quantum entanglement is when quantum objects interact so that the state of each object cannot be described independently of the other objects [8]. When a pair of quantum particles are entangled, it is possible to know the state of both particles by measuring one of them. For example, if two electrons are entangled, they can be described as being in the superposition of states spin up and spin down. If one electron is measured, then the state of that electron is correlated with the state of the other. If the measured electron is spin up, then the other one is also spin up.

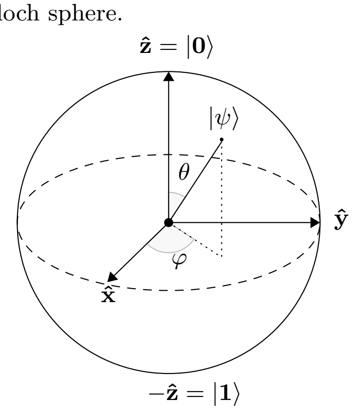
2.1.3 *Decoherence*

A system maintains its quantum state as long as the environment does not disturb it. All quantum systems lose their quantum states because they can never be perfectly isolated from their environments. Decoherence is when a quantum state is disrupted by the environment. When two-level quantum systems decohere, they go from being in a superposition of states to one state. For example, an electron can be in a superposition of spin up and spin down states for a short period of time before decohering to only being in a spin up or spin down state. The length of time it takes for a particle to lose its superposition of states is called the decoherence time.

2.2 Quantum Computing

2.2.1 *Quantum bit*

A fundamental unit of information for classical computers is the bit. The fundamental unit for quantum computers is the quantum bit, or qubit. It can be represented with a Bloch sphere.



While a bit can be represented as a vector that occupies one of two states, a qubit needs to be described as a two-level quantum system. This means that is can be represented as a vector on two dimensional complex vector space, like a Bloch sphere. Computations are performed by rotating the state vector in the sphere before the qubit decoheres [6].

2.2.2 *Quantum gates*

Like classical computing, quantum computing has finite set of universal gates. A set of universal quantum gates can be used to construct any arbitrary quantum algorithm. The most common types of universal quantum gates are one-qubit gates and two-qubit gates. One-qubit gates can be represented with 2 by 2 matrices. They transform the state vector of one qubit. 4 by 4 matrices are used to represent two qubit gates. They are used to entangle 2 qubits with each other. A set of universal quantum gates can be composed of Pauli and CNOT gates. They can also be composed of Hadamard, Phase shift, and CNOT gates [6].

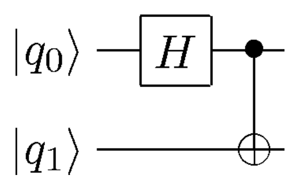
$$Pauli - X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Pauli - Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Pauli - Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\pi/8 = \begin{bmatrix} 1 & 0 \\ 0 & e^{\pi/4i} \end{bmatrix} \quad Hadamard = \frac{\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}}{\sqrt{2}}$$

2.2.3 *Quantum Algorithms*

A quantum algorithm is the application of successive quantum gates to qubits. It can be represented as multiplying the vector that represents a gate with the vector that represents a qubit. It can also be expressed as a circuit diagram.



The above circuit diagram is the Bell state algorithm. This algorithm is an illustration of quantum entanglement. The Hadamard gate is applied to the first qubit, which transforms the state of the first qubit to a superposition. The CNOT gate is applied from the first qubit to the second one. This will perform the NOT operation on the second qubit if the first one is in an excited state, and do nothing if the first qubit is in a ground state [6].

2.2.4 *Quantum error Corection*

Quantum error correction techniques are used when the quantum algorithms need to be protected from decoherence and noise; because decoherence and noise destroys the computation. Redundant qubits are used in the physical implementation of a quantum computer to preserve the state of the computation for as long as possible [6].

2.2.5 *DiVincenzo's criteria*

DiVincenzo's criteria is a list of specifications created by David DiVincenzo that a physical system must satisfy to be considered a universal quantum computer [2].

1. System of qubits
2. Method of initilizing the qubits to a ground state
3. Qubits with long decoherence times
4. Implementation of a universal set of quantum gates
5. Method of measuring state of qubits

2.2.6 Computation with Ion-traps

Ion traps can be used to meet DiVincenzo's criteria. They are devices that use magnetic and electric fields to trap ions [8]. Each ion that is trapped represents a qubit. These qubits are initialized by cooling them to their ground states. Ions are a great representation for qubits because they have long decoherence times (from hundreds to thousands of microseconds) [6].

Universal quantum gates are usually implemented by shining a global laser on the ions for multiple qubit gates and focused lasers for single qubit gates. The state of the qubit is measured by shining a laser on an ion and observing if the ion emits a photon. If the ion emits a photon, then the ion is in an excited state. If nothing is emitted, then the ion is in a ground state. The frequency for the laser that performs measurements and the one that applies quantum gates is different, because ions need to maintain their quantum state after the gates are applied.

3. PREVIOUS WORK

Most computer architectures whether classical or quantum, generally follow the same design. The architecture is composed of an instruction set. This instruction set describes a language for an abstract machine which could implemented in software or hardware.

3.1 Laser ion trapped architecture

3.1.1 Instruction Set Architecture

Except for some minor differences, most ISAs for quantum computers are the same. The high level ISA is represented as the text representation of the quantum gates. An ISA for a laser trapped ion computer could be the based on the Von Neumann model [5].

3.1.2 Physical implementation

This computer is composed of several ion traps that are connected with wires. Computations are performed by applying lasers on the ions. The wires are used to transport ions between traps.

3.2 Quil

A hybrid quantum computer is a computer that treats a quantum computer as a co-processor to a classical computer. It does not give the quantum computer a job that cannot be performed efficiently on the classical computer. Near-term quantum computers will be hybrid. One instruction set for hybrid quantum computers is Quil. This instruction language describes an abstract machine for a hybrid quantum computer. It is primarily designed for near-term algorithms; like the variational quantum eigensolver and material simulations [7].

3.3 Microwave ion trap quantum computer

3.3.1 Physical implementation

The trapped ion microwave quantum computer is made out of individual modules. Each module is a universal quantum computer. It contains a loading zone, gate zone and readout zone. The loading zone is where ions are initialized. Quantum gates are applied to the ions in the gate zone. Instead of using lasers to perform the gates like the laser trapped ion computer, this computer performs gates using microwave radiation. The readout zone is where a measurement laser is applied to the ion to read the result of the computation. Individual modules are attached to each other to create a large scale quantum computer [5].

3.3.2 Error correction

The trapped ion computer uses surface codes to perform error correction [5]. The physical qubits are separated into data qubits, measure-x and measure-z qubits. The measure-z qubit applies 4 CNOT gates on the data qubit. The measure-x performs the Hadamard gate before and after the 4 CNOT gates have been applied. It is designed so that if there is an error in a data bit, it flips the neighboring operators that will detect the error. The data for these errors will be transferred to a classical computer, where a classical program will correct the results of the quantum computation based on the error information.

4. NEW ARCHITECTURE

4.1 Quil compiler

I propose a new architecture that will implement the Quil ISA on the microwave ion trapped quantum computer. A lot of architectures have been proposed for ion trap quantum computers. This design borrows ideas from Balensiefer, Kregor-Stickles, and Oskin's infrastructure; which is composed of a source compiler, an error correction compiler, a device scheduler, and a simulator [1]. The source compiler will generate Quil code. Since Quil code is a low-level intermediate representation of quantum programs [7], The Quil compiler will be a back-end compiler. It will contain 3 phases. Program analysis, optimization, and code generation. The code generator will output an architecture-dependent ISA. This architecture-dependent ISA will have a one-to-one correspondence with the machine code that will be executed on the computer.

4.2 Error correcting compiler

Error correction will not be needed for near-term quantum devices. When quantum computers are composed of enough qubits, we will be capable of programming them to perform error corrected computations. This is why the error correcting compiler is optional. Unlike the Quil compiler, the computer programmer can choose to incorporate the error correction compiler into his/her architecture. The error correcting compiler will be a source to source compiler. It will compile Quil code to error corrected Quil code. If the computer architect wishes to incorporate the error correction compiler into his/her architecture, it will be placed before the Quil compiler. Its input will be the Quil code that that was written by the programmer or code generated by a higher level compiler, and it's output will be the error corrected version of that program. The error corrected code will be valid Quil code. This error corrected program will be the input to the Quil compiler, which will generate the architecture-dependent ISA. The error correction compiler will implement the surface code error correction scheme that was covered in section 3.3.2.

5. BENEFITS OF THE NEW DESIGN

Quil is a low-level intermediate representation. This means that it can be treated like the LLVM Intermediate representation to build reusable compilers [4]. Compiler writers who choose Quil as a target language do not have to write a back-end for their compiler. Using Quil as the implementatation-independant ISA for the microwave computer takes advantage of this reusability.

The modularity of the architecture means that other architecture designers can take the parts they like about this implementation and leave parts they would like to implement on their own. If they like the error correction compiler, but their implementation targets a different back-end like a superconducting quantum computer, they can take the error correction compiler and implement everything else. They could also decide to add their backend to the Quil compiler.

6. CONCLUSION

I have proposed a scalable software architecture for a scalable quantum computer. After reviewing the available tools, I have found that they were not sufficient for near-term and long-term ion trap quantum computers. The Quil ISA is designed for near-term quantum computers because it compiles straight to the implementation of an abstract machine without first going through an error correction compiler. If the Quil ISA is compiled to an error correcting ISA, it can be suitable for long-term quantum computers. The ideas presented are mainly theoretical, but there is a plan to test them in future research.

7. FUTURE WORK

The ideas presented in this paper are not complete; specifically, how the Quil and error correction compilers will be implemented. This paper does not go into the detail about the algorithms and data structures these compilers will use. It also doesn't talk about the design of the architecture-dependent ISA. Specifications of the compilers and the architecture dependent ISA might be presented in future papers.

The difference between a virtual machine and a simulator is that a virtual machine is an implementation of an abstract machine, while a simulator is a software implementation of a specific hardware. An open source simulator for near-term hybrid quantum computers does not exist. Most of the open source simulators are not practical for running near-tem quantum algorithms like the variational quantum eigensolver. If not incomplete, they are usually for pedagogical or research purposes. A possible direction for the future would be implementing an open source virtual machine for near-term devices, and a wide range of programming tools on top of this virtual machine.

Another possible direction is the implementation of an open source simulator of the microwave trapped ion quantum computer. This is because there is no way to know how the sound the ideas presented in this paper are until they are implemented in software. This simulator will be used for testing purposes, like how decoherence affects the computation, and how error correcting algorithms mitigate these effects. It will also test the viability of the microwave quantum computer by testing how well the computer modules scale with respect to the number of qubits. This simulator might follow design of the Monte-Carlo simulator [1].

8. ACKNOWLEDGMENTS

I would like to thank Joseph Zambreno for providing valuable feedback on how to study this subject. I would also like to thank Lucas Skoff, whose friendship kept me sane while I was performing this research.

9. REFERENCES

[1] S. Balensiefer, L. Kregor-Stickles, and M. Oskin. An evaluation framework and instruction set architecture for ion-trap based quantum micro-architectures. In *Proceedings of the 32nd annual international symposium on Computer Architecture*, pages 186–196, New York, NY, USA, 2005. ACM.

[2] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000.

[3] R. Feynman, S. Sands, Matthew, and R. Leighton. *The Feynman Lectures on Physics Volume 3: QUantum Mechanics*. Basic Books, 250 West 57th Street, 15th Floor, New York, NY,10107, USA, 1965.

[4] C. Lattner and V. Adve. LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation. In *Proceedings of the 2004 International Symposium on Code Generation and Optimization (CGO'04)*, Palo Alto, California, Mar 2004.

[5] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger. Blueprint for a microwave trapped ion quantum computer. *Science Advances*, 3, July 2017.

[6] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, New York, NY, USA, 2010.

[7] R. S. Smith, M. J. Curtis, and W. J. Zeng. A practical quantum instruction set architecture, 2016.

[8] P. Wieburg. A Linear Paul Trap for Ytterbium Ions. Master's thesis, Universität Hamburg, 20146 Hamburg, Germany, 2014.