EXPERIMENTS IN QUANTUM COMPUTING

Peter Jones, Department of Engineering Science, University of Greenwich
Charles A Nicholls, Department of Engineering Science, University of Greenwich
Shanker G R Prabhu*, Department of Engineering Science, University of Greenwich
s.prabhu@gre.ac.uk*

Keywords: Quantum Computing, Game Theory, Grover’s Algorithm, Quantum Cloning, Coin Tossing

Abstract

Quantum Computing is widely perceived to be one of the ways forward in the future of computation as the end of *Moore’s law* is almost here. Instead of using bits in classical computers, quantum computers manipulate qubits which are governed by the phenomena of superposition and entanglement. In this paper, we demonstrate the relevance of quantum computing in game theory and database search applications. Through a simple example of coin tossing, we show how it is possible to organise a game where one player adopting a quantum strategy is guaranteed to win. It is also shown that the other player, using a second coin (qubit), can subvert this action to award the wins to themselves without the first player’s knowledge. In another application, a modified Grover’s database search algorithm is applied to clone an arbitrary quantum state of a qubit to a duplicate qubit. In both cases, the comparison of simulated and actual results emphasises on the hardware limitations of the current error-prone quantum computers. The quantum computer programs are designed using quantum gates and simulated in the Quantum Information Software Kit before testing on the IBM Q 5.1 (*ibmqx4*) quantum computer.

1. INTRODUCTION

The concept of Quantum computing was first introduced in the 1980s by Feynman where it was hypothesised that computation can be performed in the quantum space [1]. Even though the theory is decades old, the field received renewed attention since the past five years due to various reasons such as being able to physically build quantum computers and due to the quest to move away from stagnating classical computing technology. The transistor based classical computing systems have been constantly subjected to miniaturisation to keep up with *Moore’s law* (or observation) for almost five decades. However, the current technology has reached a stage where it is impractical to further miniaturise computing chips as principles of classical physics is no longer applicable and quantum theory has taken over [2]. This draws the attention towards exploring quantum computing as a viable alternate solution.

The basic unit of data in Quantum Computing is a qubit or quantum bit with two states represented in *Dirac* notation (also called Bra-Ket notation) as $|0\rangle$ and $|1\rangle$ vector states. The field operates using the effects of quantum entanglement and superposition [2]. Quantum entanglement is when two or more quantum states become determined by the state of one another. Hence, if one state is manipulated, then the other state is also simultaneously manipulated. Quantum superposition allows for more than one quantum state to be added together for the creation of another state. This means a $|1\rangle$ state can be superimposed with a $|0\rangle$ state for the creation of a new state of binary superposition shown in *Equation 1*,

$$|\psi\rangle = \frac{|1\rangle+|0\rangle}{\sqrt{2}} \qquad \textbf{Equation 1}$$

A *Bloch* Sphere is used to graphically visualise a qubit with the vector pointing up for $|0\rangle$ and down for $|1\rangle$ (*Fig 1*). The applications explained in this paper are designed for the cloud accessible 5 qubit IBM Q 5.1 (*ibmqx4*) quantum computer. The Quantum Information Software Kit is used to program the computer with quantum gates in quantum circuits, called quantum scores in the quantum composer. The quantum scores are initially simulated in a local machine before testing them on the IBM Q machine [3].

2. APPLICATION IN GAME THEORY

This paper builds on the application of a quantum computer in a scenario of coin tossing as a game presented by Meyer [4]. The game rules are as follows: Each of the two players, Picard and Q are allowed to flip a coin placed in a box one after the other. The coin initially is placed heads-up by Picard and players are not allowed to see each other's moves. At the end of two turns, Q wins the game if the coin remains heads-up. The game moves shown as a state flow is depicted in Fig 1.

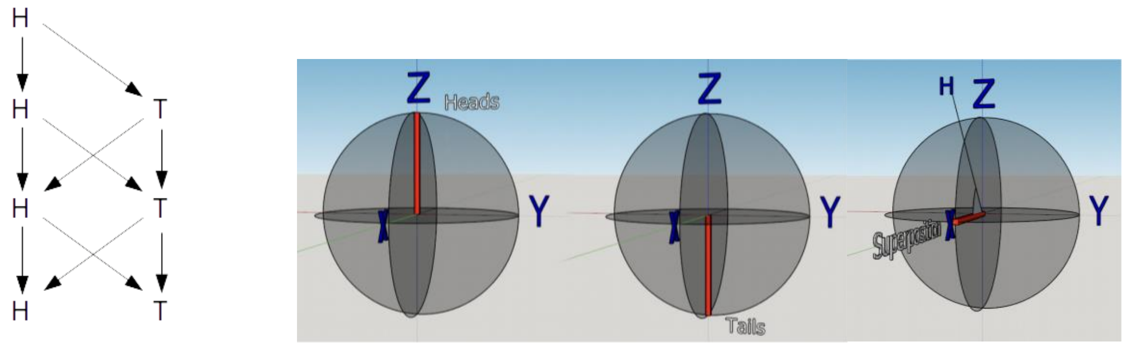

Fig 1. Game Moves (H= Heads and T= Tails) and corresponding quantum states with superposition in a Bloch sphere.

Even though [4] shows how Q using a quantum strategy can cheat to always win the game, it does not explain how Picard can win back. Therefore, the paper presents the alternate scenario where Picard is guaranteed to win along with a scenario to balance the odds for both players.

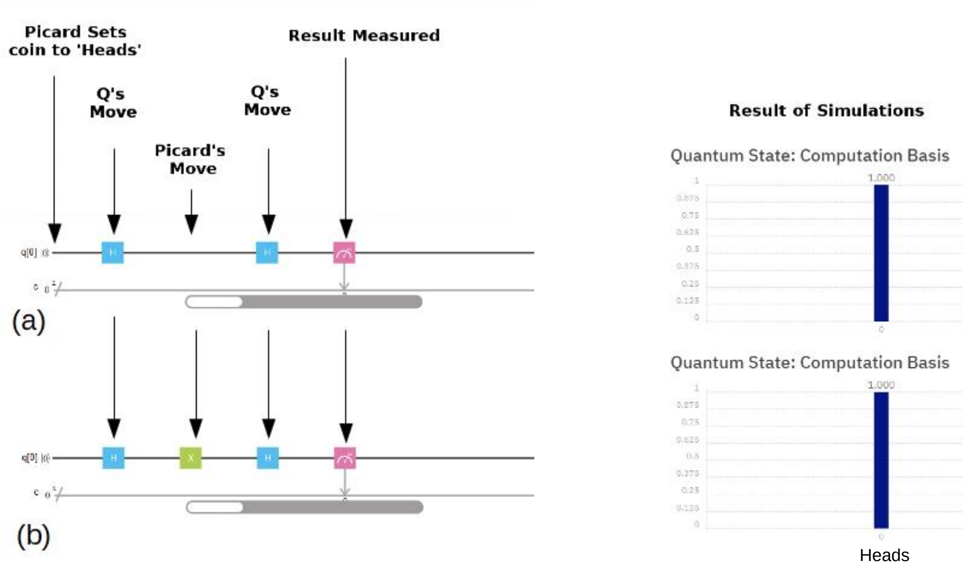

Fig 2. Scenario where Q wins. (a) and (b) are possible combinations with the probability of observing Heads at the output. Simulations with both combinations show 100% probability of Heads at the end of the game. q[0] is the qubit used for tossing, H gate is in blue and X gate is in green.

2.1 Implementation

2.1.1 Scenario 1: Q wins

In the scenario of Q always winning the game, during the first turn, the vector is rotated with an *Hadamard (H)* gate to bring it to a state of superposition as explained in *Equation 1* and *Fig 1*. This is effectively rotating the vector by π radians along the *X-axis* and $\pi/2$ radians along *Y-axis*. Later, irrespective of the moves from Picard in the final turn, Q can reverse the rotations to bring it back to the original heads-up state with another *H* gate. The Quantum Scores with simulation results for the possible combinations in this scenario are shown in *Fig 2*. The coin flip is accomplished by an *X* gate (in green). From simulation results in *Fig 2*, it is evident that Picard's move does not impact the final result.

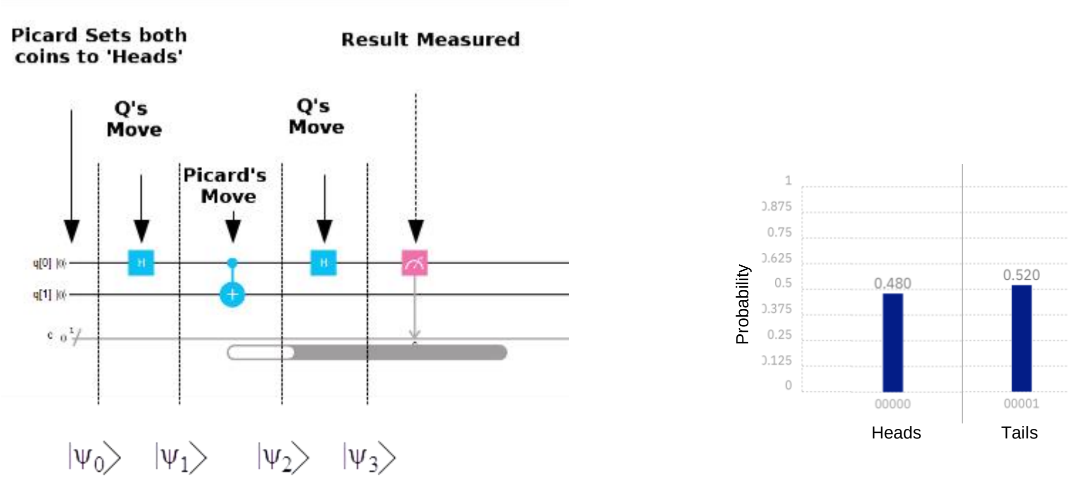

Fig 3. Scenario to balance the odds of the game. The measured output probabilities show ~50% chance for Heads and Tails. q[1] is additional qubit introduced by Picard to entangle the first qubit (q[0]).

2.1.2 Scenario 2: Fair play

To ensure equal probabilities for both players to win the game, an entangled qubit (or coin) through a *Controlled NOT (CNOT)* gate can be introduced in one of the turns by Picard as shown in *Fig 3 (*between $|\psi_1\rangle$ & $|\psi_2\rangle$). The operations in *Fig. 3* can be mathematically explained as follows:
At the start $|\psi_0\rangle = |00\rangle$. Each state is therefore given by:

$$|\psi_1\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)(|0\rangle) = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle \qquad \textbf{Equation 2}$$

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle\right) \qquad \textbf{Equation 3}$$

$$|\psi_3\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\frac{1}{\sqrt{2}}|0\rangle \ + \ \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\frac{1}{\sqrt{2}}|1\rangle \ = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle \qquad \textbf{Equation 4}$$

Since in the game, Q is only aware of the first qubit, the outcome at measurement ($|\psi_3\rangle$) has a probability 0.5 in both *Heads* and *Tails* cases. This is verified by the simulation results in *Fig 3*.

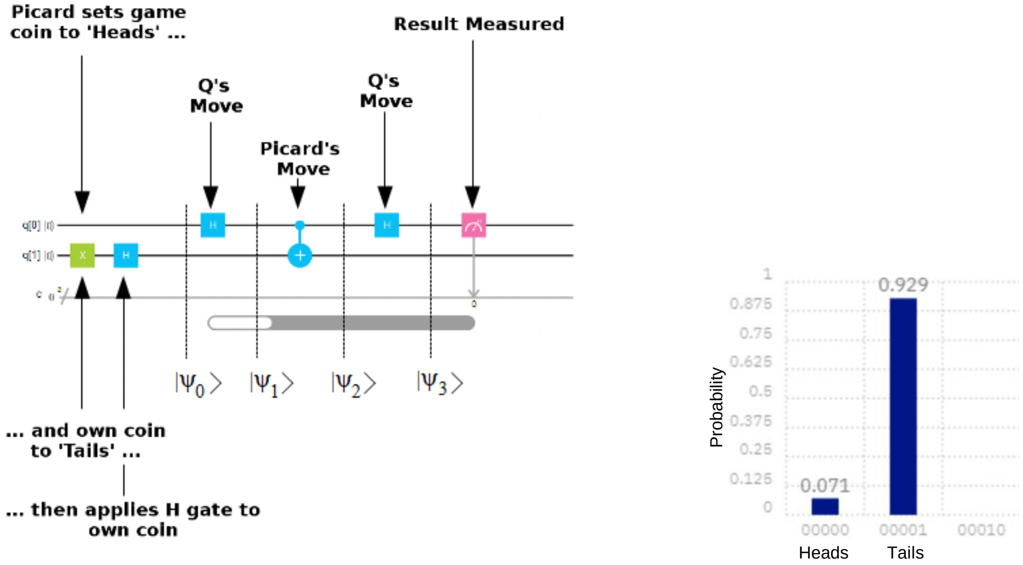

Fig 4. Scenario where Picard always wins. The actual measured output probabilities show 92.9% chance for Tails instead of an expected 100%. q[1] is initialised as Tails with an X gate and H gate is introduced to bring it in a state of superposition.

2.1.3 Scenario 3: Picard wins

In order for Picard to guarantee a win after each game, the game coin should always turn *Tails.* This can be performed by entangling the game coin or qubit to a known initial state. The process shown in *Scenario 2* is repeated with a difference of second qubit initialised as *Tails* and an *H* gate applied to it. The new quantum score with simulation and actual results are shown in Fig 4. The effect of operations performed on the intermediate states are:

$$|\psi_0\rangle = (|0\rangle)\left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right) \qquad \textbf{Equation 5}$$

$$|\psi_1\rangle = \left(\tfrac{1}{\sqrt{2}}|0\rangle + \tfrac{1}{\sqrt{2}}|1\rangle\right)\left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right) = \tfrac{1}{2}|00\rangle - \tfrac{1}{2}|01\rangle + \tfrac{1}{2}|10\rangle - \tfrac{1}{2}|11\rangle \qquad \textbf{Equation 6}$$

$$|\psi_2\rangle = \tfrac{1}{2}|00\rangle - \tfrac{1}{2}|01\rangle + \tfrac{1}{2}|11\rangle - \tfrac{1}{2}|10\rangle = \left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right)\left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right) \qquad \textbf{Equation 7}$$

$$|\psi_3\rangle = |1\rangle\left(\tfrac{1}{\sqrt{2}}|0\rangle - \tfrac{1}{\sqrt{2}}|1\rangle\right) \qquad \textbf{Equation 8}$$

The expected output of the this scenario is the bit q[0] appearing as 1 (or *Tails*) with 100% certainty, but *Fig 4* shows the result as having only 92.9% certainty due to various error sources in the machine.

2.2 Conclusion

When playing a simple game like coin flipping, it has been shown that one player can gain an advantage by using a quantum strategy instead of adopting the classical strategy of just using two states (*Heads* or *Tails*). If both players use quantum strategies an individual advantage may be gained only as long as the other party thinks that the opponent's strategy is classical.

3. QUANTUM CLONING

The no clone theorem in quantum mechanics states that, when applied to qubits, there is no possible method to reproduce multiple qubits from an initial unknown qubit [5]. The theory is often seen as a double-edged sword for quantum computers as it both inhibits the ability to perform data integrity checks on qubits inside a program, but also allows for secure data transfer as no accurate copies of the data can be made. Whilst the no clone theorem makes accurate repeatable copying of a qubit's quantum state impossible it does allow for the potential for inaccurate or imperfect clones of a quantum state to be made. The construction of these imperfect clones is the objective of the algorithm outlined in this paper.

3.1 Algorithm

Grover's algorithm is a well-known quantum algorithm used for unstructured database search [7]. The data to be searched is embedded in the quantum oracle function (U_w) which returns 1 if the search returns a valid solution or else 0. The U_s function then amplifies the amplitude of the search item and the output of the system then has an increased probability that the measured result will match the search term. We use this property of the algorithm to copy two qubits on to another two blank qubits. As shown in *Fig 5*, *CNOT* gates entangle input qubits $|1\rangle$ or $|0\rangle$ (q[1] and q[3]) into the oracle function. The figure also shows the algorithm and its various stages moving from qubit input to Grover's algorithm to the measurement stage. q[1] and q[3] are inputs with output measured at q[0] and q[2].

3.2 Results and Discussion

Experiments were conducted with all four two qubit input combinations and for states of superposition in both simulation for 100 times and on IBM Q 5.1. The simulated results in *Table 1* indicate 100% repeatability of experiments where there was a 100% chance of observing the right output of copying the input qubits. However, as seen from *Table 1*, when the same was run 1024 times on actual machine, the probability of obtaining a perfect clone was less than 15%. The results for input qubits in superposition in *Fig 6* shows that although the output qubit always matches the input qubit they are not clones. Or, if they are clones of the input qubit then they would have their own probability of being a $|1\rangle$ or $|0\rangle$ and not be entangled with the input qubits. The possible observed states would then be $|00\rangle, |01\rangle, |10\rangle$ and $|11\rangle$ rather than just the two states $|00\rangle$ and $|11\rangle$ observed in the simulation results.

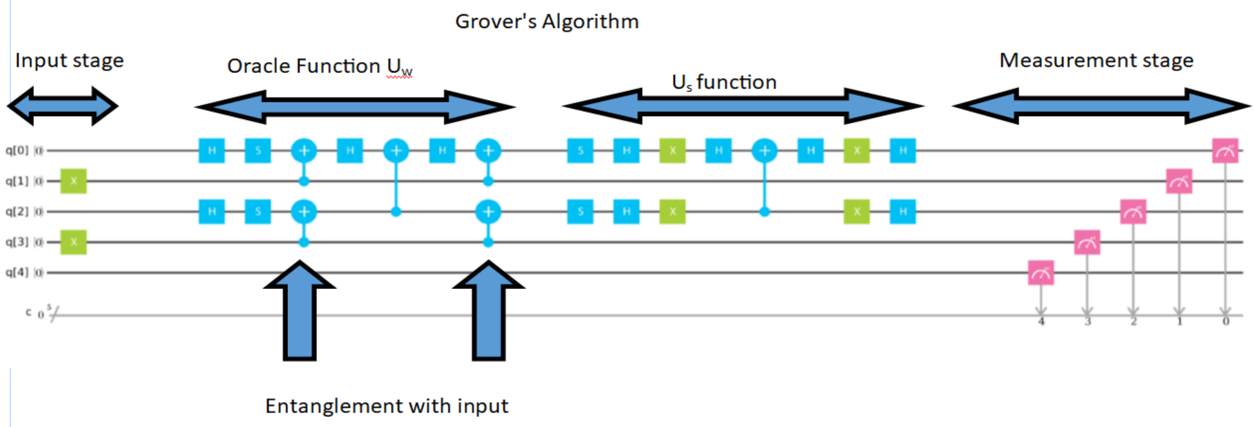

Fig 5. Grover's Algorithm with input at q[1] and q[3] and output at q[0] and q[2].

Table 1. Performance of the algorithm while cloning.

Input	Expected Output	Simulated	Actual			
		Chance of perfect match	Chance of q[1] match	Chance of q[3] match	Chance of input error	Chance of perfect match
|11⟩	|1111⟩	100%	63%	89%	41%	11%
|10⟩	|1100⟩	100%	64%	73%	38%	13%
|01⟩	|0011⟩	100%	66%	80%	42%	10%
|00⟩	|0000⟩	100%	62%	81%	47%	12%

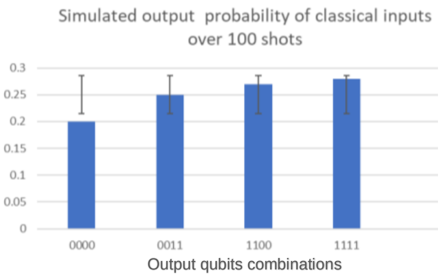

Fig 6. Simulated probabilities for input qubits while in superposition.

3.3 Conclusion

Finding a more accurate way of copying the state of a qubit would allow for data integrity checks within a quantum algorithm. The probability of a match using this algorithm is very low for q[1] at 60-65%, is only a small improvement on 50% random chance but with q[3] the rate is higher at a 75-80%. They are not equal as is not currently possible to run the algorithm in parallel as there is still no way to accurately copy a qubit into multiple parallel inputs and due to multiple error sources during computation.

4. ERROR SOURCES

Any physical implementation of a Quantum Computer must deal with effects that conspire to prevent information being stored reliably at these tiny scales. Called quantum decoherence, interactions between the quantum state carriers and their environment introduces random factors into the system and inhibits the carriers' ability to interfere with each other. Decoherence takes two major forms: 1) Dephasing and 2) Energy Relaxation. Dephasing occurs when the correlation of phase within the quantum system breaks down and becomes randomised. Energy Relaxation is the decay of the excited |1⟩ state towards the ground |0⟩ state. Both of these undesirable reactions are mitigated in part to cooling the system to as close to zero Kelvins as possible [7].

When measuring the ability of a quantum computer to remain in a coherent state over time, both of these effects are combined into a figure-of-merit that takes the form of a time-constant *T2*. At the time of conducting experiments reported in this paper, the best *T2* attainable is of the order *100 μS*, although for the experiment of *Fig. 7*, the *T2* for each gate of the *ibmqx4* was a just under *50 μS* [3]. The longer *T2*, the better the ability to retain quantum coherence.

Experiments are run on the *ibmqx4* as *1024* individual shots in order to reduce the effects of decoherence as much as possible. At the end of the experiment, all of the shots are averaged to give a probabilistically reliable indication of the result. The expected output in *Section 2.1.3*, is the qubit q[0] appearing as 1 with 100% certainty, but *Fig. 7* shows the result as having only 92.9% certainty. This is the due of the decoherence mechanisms described. In general, the longer the Quantum Score, the greater the decoherence encountered.

There are other possible error sources that are accountable for the large differences in measured and simulated probabilities. Gate errors and readout errors for *ibmqx4* amass over the 15 gates in the cloning algorithm and can account for an error of 10% on the input qubits and 20% on the output qubits in *Section 3*. It implies that if the gate errors could be further improved, a success rate of around 95% could be expected from q[3] (*Section 3*). This would increase the usefulness of the cloning algorithm to a state where it would have a meaningful output.

5. CONCLUSION

Even though practical quantum computers are expected to be available in the next five years [8], real-world quantum computing is still in its infancy and the deleterious effects of environmental interaction mean that it is a struggle for the designers to keep the qubits isolated from confounding factors. At the moment, only cutting- edge techniques suffice in cooling the computers down to *15 mK* above absolute zero, but as technologies progress, better techniques will make quantum computing more easily accessible. Further, in applications proposed in the paper, a quantum strategy can definitely be used for coin tossing. But the error from cloning algorithm is unacceptable for any practical use. However, with some work on error reduction, the future applications of the cloning algorithm could include date integrity checks or in cryptography. If the probability of success was improved, a qubit could be copied on to multiple parallel lines and then the outputs could be compared to reduce gate errors. This would allow gate errors to be reduced or even eliminated during the processing of a qubit.

6. REFERENCES

[1] R. P. Feynman, “Simulating physics with computers,” *International journal of theoretical physics*, vol. 21, no. 6-7, pp. 467-488, 1982.

[2] L. Gomes, “Quantum computing: Both here and not here,” *IEEE Spectrum ,* vol. 55, no. 4, pp. 42-47, 2018.

[3] IBM, “ibm-q,” 2018. [Online]. Available: https://www.research.ibm.com/ibm-q/. [Accessed 22 March 2018].

[4] D. A. Meyer, “Quantum strategies,” *Physical Review Letters*, vol. 82, no. 5, pp. 1052–1055, 1999.

[5] J. L. Park, “The concept of transition in quantum mechanics,” *Foundations of Physics,* vol. 1, no. 1, pp. 23-33, 1970.

[6] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996.

[7] B. E. Kane, “Silicon-based quantum computation,” *Fortschritte der Physik,* vol. 48, no. 9–11, pp. 1023–1041, 2000.

[8] R. Juskalian, “Practical quantum computers”, *MIT Technology Review,* 2017. [Online]. Available: https://www.technologyreview.com/s/603495/10-breakthrough-technologies-2017-practical-quantum-computers/. [Accessed 10 May 2018].