

QUANTUM COMPUTING

ELLIOTT ASHBY
PHYSICS AND ASTRONOMY
UNIVERSITY OF SOUTHAMPTON

ABSTRACT. This report provides a high level overview of quantum computing. The history and development during the 20th century are discussed as well as the fundamental principals underpinning quantum computing such as qubits, superposition and entanglement. The report also discusses implementations of quantum computing such as quantum circuits and algorithms as well as the reasoning behind why those algorithms are needed to continue the current pace in the speed up of computation and even simulate physics impossible on a classical computer. Finally, the report discusses the criteria and requirements to build a quantum computer.

CONTENTS

1. Introduction	2
2. An Overview of Key Concepts	2
2.1. A History of Quantum Computing	2
2.2. Limitations of Classical Computers and the Need for Quantum Computing	3
2.2.1. Public-key Cryptography and Factorization of Big Numbers	3
2.2.2. Brute-force Search	4
2.2.3. Simulation of Quantum Systems	4
2.3. Quantum Bits and Superposition	5
2.4. Quantum Superposition of Multiple Qubits and Entanglement	6
3. Quantum Circuits	7
3.1. Quantum Gates	7
3.2. Copying a Qubit	11
4. Quantum Algorithms and Parallelism	12
4.1. Deutsch-Jozsa Algorithm	12
4.2. Bernstein-Vazirani Algorithm	13
4.3. Shor's Algorithm	14
5. Experimental Quantum Computing	16
6. Conclusion	16
Appendix A.	
Additional Information	17
Appendix B.	
Quantum Fourier Transform	17
Appendix C.	
Further Reading	19
References	19

Date: January 2025.

1. INTRODUCTION

Animals gain advantages in many ways, one of which is the exploitation of properties of the physical world. This has come to culmination in humans; in 1941 we saw the creation of the first programmable computer, the Z3, by Konrad Zuse, and in the following decades we have continually perfected this technology. The modern computer that we use today is, at its fundamental principals, identical to the Z3, performing binary operations on "bits" (a 1 or a 0) of data in order to encode useful computational results.

The Z3 used "electromagnetic relays (600 in the arithmetic unit, 1,400 to store 64 words)" and was "close in size to the Stibitz BTL Model 1" or a large, floor-to-ceiling bookshelf. [1] In the decades following the Z3, the space required to store and operate computer memory has decreased significantly, and as stated by Moore in 1965, "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year." [2] This has largely held true, thanks to increasingly smaller and smaller manufacturing processes.

As of 2022 the smallest transistors are of the order of 3nm, [3] but when we shrink further down to 2nm or beyond, we begin to approach the size of the atom; at this scale, quantum effects are more pronounced and transistors can leak current due to gate direct tunnelling. [4] These effects limit the effectiveness of classical computers at this scale, and so we must look to new technologies to push the boundaries of computation.

2. AN OVERVIEW OF KEY CONCEPTS

2.1. A HISTORY OF QUANTUM COMPUTING.

During the majority of the 20th century up until the early 1980s the fields of quantum mechanics and computer science were, for the most part, separate areas of study despite some crossover such as the laser. But in 1980, Paul Benioff proposed a quantum mechanical model of the computer and computation process [5] and additionally, in the same year, Yuri Manin proposed a similar model. [6] In the following years, Richard Feynman wrote a paper suggesting that the use of quantum phenomena to perform computations could be more efficient for computer physics simulations than classical computers. [7] 2 years later, in 1984, Charles Bennett and Gilles Brassard continued to merge quantum mechanics and computer science by introducing quantum cryptography [8] showing that a quantum key distribution can be used to secure communications.

Following the proposal of the quantum model of computation, quantum algorithms began to be developed, including Deutsch's algorithm in 1985, [9] the Bernstein-Vazirani algorithm in 1993, [10] and Simon's algorithm in 1994. [11] Building on these papers, Peter Shor published his work on prime factorization quantum algorithms which had real world application breaking the RSA and Diffie-Hellman encryption algorithms. [12] By 1996, Lov Grover published his algorithm for database search [13] and in the same year Seth Lloyd finally proved Feynman's conjecture that he proposed in 1982 that quantum computers (QCs)

”can be programmed to simulate any local quantum system”. [14]

The first QC to be built was in 1998; Isaac Chuang and Neil Gershenfeld along with Mark Kubinec implemented Grover’s search algorithm with a 2-quantum bit (qubit) nuclear magnetic resonance (NMR) QC. [15] In the years following, NMR QCs would increase in the number of qubits allowing for a 7-qubit QC to run Shor’s algorithm in 2001. [16]

Since then, quantum computing has continued to grow, with new technologies and greater numbers of qubits. The most promising of these is the superconducting circuit used as a qubit. Originally proposed in 1999 by Yasunobu Nakamura, Yuri Pashkin and Jaw-Shen Tsai, [17] and shown to be viable for greater application in 2007 by Jelle Plantenberg, P.C. de Groot, C.J.P.M. Harmans and Hans Mooij by demonstrating the controlled-NOT gate, [18] superconducting qubits have become the focus of many large companies such as IBM and Google.

As of 2024, the QC with the most number of qubits is actually not a superconducting QC, but an atomic array QC built by Atom Computing in 2023 [19, 20] with a reported 1,180 qubits. However, since atomic array QCs are much newer than superconducting QCs, they have less general support for quantum algorithms; the largest superconducting QC as of 2024 is IBM’s Condor with 1121 qubits. [21, 22]

While quantum computing is still in its infancy, the promising results and fast pace of its evolution means that QCs are highly likely to become more relevant and more mainstream in the near future.

2.2. LIMITATIONS OF CLASSICAL COMPUTERS AND THE NEED FOR QUANTUM COMPUTING.

2.2.1. *Public-key Cryptography and Factorization of Big Numbers.*

In 1976, Whitfield Diffie and Martin Hellman published their paper on new directions in cryptography, [23] introducing to the general public the concept of public-key cryptography; a method of encryption that uses a pair of keys, public and private, to encrypt and decrypt messages. With the public key, one can encrypt a message that only the private key can decrypt. 2 years later in 1978, Ron Rivest, Adi Shamir and Leonard Adleman published their paper on the ”RSA” algorithm, named after their initials, that provides a method for generation of the keys. [24]

The problem of breaking the RSA algorithm is one of finding the prime factors of a large integer. This is difficult to solve for classical computers; the best known algorithm, the general number field sieve (GNFS) [25] runs in sub-exponential time, with a time complexity of:

$$O\left(\exp\left[\left(\frac{64}{9}\right)^{1/3}(\log n)^{1/3}(\log \log n)^{2/3}\right]\right)$$

where n is the number to be factorized. If we were to use an n of 2^{2048} , we would yield a number of operations roughly 1.53×10^{35} . On a 5GHz CPU on a single thread, this would take 9.72×10^{17} years, a long enough time that the RSA algorithm would be considered secure.

However in 1994, Peter Shor published his quantum algorithm for prime factorization [12] now known as Shor's Algorithm. Shor suggests that a QC could be able to factorize a number in polynomial time $O(\log n)$. The fastest current implementation is of $O((\log n)^2(\log \log n))$ [26, 27] yielding a number of operations 1.46×10^7 . This is a significant speed improvement over the GNFS and implies that the RSA algorithm is no longer secure. Among experts, security through obscurity is not considered a valid form of security, [28] and as such, the need for further research into quantum computing and quantum-resistant cryptography is clear. Shor's Algorithm is discussed further in section 4.3.

2.2.2. Brute-force Search.

Brute-force search systematically generates and tests all potential solutions to a problem. That is to say, brute-force search find the single item that satisfies some condition in a unsorted database of n items. Once a single item has been examined, its ability to satisfy the condition can be determined in one step and as such, the most efficient classical algorithms may only determine the correct item in $O(n)$ time, averaging $n/2$ operations.

In 1996, Lov Grover published his quantum algorithm for database search [13] now known as Grover's Algorithm. Grover's Algorithm is able to find this single item in only $O(\sqrt{n})$ steps, and although not a polynomial time speed up, it is still a great increase in speed over classical algorithms. Grover's Algorithm is possible since quantum mechanical systems can be in a superposition of states and simultaneously evaluate the conditions for multiple items in the database.

A year later in 1997, Grover's Algorithm was shown to be asymptotically optimal by Charles Bennett, Ethan Bernstein, Gilles Brassard and Umesh Vazirani. [29] An algorithm can be called asymptotically optimal if it performs, at worst, a constant factor slower than the best possible algorithm for large inputs.

2.2.3. Simulation of Quantum Systems.

Can physics be simulated on a classical computer? In 1982, Richard Feynman states that it is definitely impossible to simulate quantum systems on a classical computer without infinite time. [7]

If we wish to simulate a single particle, ψ as a function of x and t , its probability density can be determined classically using numerical methods. [30] However, if we wish to simulate n particles, the new state of the system is given by some function $\Psi(x_1, x_2, \dots, x_n, t)$. Describing all of these states would require a " k -digit number for every configuration of the system, for every arrangement of the $[n]$ values of x ". Then, if there are N points in space and each point in space has its own information such as electric fields, n is of order N , so there would be N^N configurations. Since there are too many variables, it cannot be simulated with a classical computer; there are more variables than estimated atoms in the

universe! Computing this classically would require to discretize x and t to make any results exact. To do this requires the discarding of terms that are too small, for example if we choose to only take k digits of precision, we must discard probabilities that are less than 2^{-k} . This is not a problem for a small number of particles, but as the number of particles increases to n , the more terms we discard, no matter their validity.

Feynman presents that discretizing presents some problems, for example taking the electric field at some point below a certain amount, would in turn suggest that it is not there at all. This is not the case, we know it to be quantized. By discretizing, we are not simulating the correct equations that reflect reality.

This leaves only one option, to simulate quantum systems on a QC. Feynman's conjecture was backed up by Seth Lloyd later in 1996 [14] stating that a mere 30 or 40 qubits would be enough to simulate a quantum simulations of multidimensional fermionic systems like the Hubbard model [31] that prove resistant to conventional computers.

2.3. QUANTUM BITS AND SUPERPOSITION.

The bit is the smallest building block of information of classical information. Similarly, the quantum bit or qubit is the equivalent for quantum information theory. [32] While the qubit can be represented by physical systems such as photons, electrons or atoms, for now we will consider the qubit as a abstract concept.

While a classical bit has a state of either 0 or 1, qubits similarly have a state of $|0\rangle$ or $|1\rangle$ which correspond to the classical states respectively. There is one key difference however, instead of being exclusively in one state or the other, a qubit can also be in a *linear combination* of states also known as a *superposition*. That is to say, a qubit can be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2.1)$$

where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. These states however, are not observable, to retrieve information from a qubit, it must be measured. Doing so we get either 0 or 1 with a probability of $|\alpha|^2$ and $|\beta|^2$ respectively. An example of this could be for a qubit to be in the following states:

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (2.2)$$

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \quad (2.3)$$

which we have denoted as $|+\rangle$ and $|-\rangle$.

If we wish to, we can also represent a single qubit using a *Bloch sphere* [33]; a sphere with radius 1 and the poles representing the 2 states $|0\rangle$ and $|1\rangle$. We can rewrite Equation (2.1) as:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \quad (2.4)$$

where θ and ϕ are the angles of the Bloch sphere shown in Figure 2.1. Operations on single qubits can be shown as rotations on the surface of the Bloch sphere, which is a useful way to visualize quantum operations.

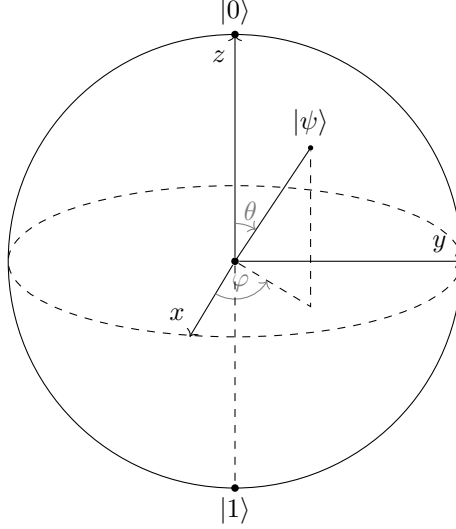


FIGURE 2.1. The Bloch Sphere.

Despite the fact that a single qubit has an infinite number of possible states (all possible values of α and β), we cannot use a qubit to store any amount of data. This is because in order to extract information from a qubit it must be measured; measurement results in a collapse from its superposition of $|0\rangle$ and $|1\rangle$ to either 0 or 1. This means that any further measurements are guaranteed to yield the same result. For example, if measurement of $|-\rangle$ gives 1, then any additional measurements must also give 1.

2.4. QUANTUM SUPERPOSITION OF MULTIPLE QUBITS AND ENTANGLEMENT.

In the same way a single qubit can be in a superposition of states, multiple qubits can be in a superposition of states. For example, a 2-qubit system can be in a superposition of 4 states:

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \quad (2.5)$$

where α_{00} , α_{01} , α_{10} and α_{11} are complex numbers and $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$. Similarly to a single qubit, the probability of measuring a state $|xy\rangle$ is $|\alpha_{xy}|^2$. However we could potentially only measure one of the two qubits, collapsing that one but not the other. If we measured the first qubit to be 1, the new state of the system would then be:

$$|\psi'\rangle = \frac{\alpha_{10} |10\rangle + \alpha_{11} |11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}} \quad (2.6)$$

But this raises a strange point, as pointed out by Albert Einstein, Boris Podolsky and Nathan Rosen in 1935 [34] that the measurement of one qubit can determine the state of another qubit simultaneously. Einstein, Podolsky and Rosen suggested that this was a problem with quantum mechanics, saying that, "No reasonable definition of reality could be expected to permit this." This is known as the *EPR-paradox*. However, later in 1964 John Bell suggested that, using a refinement on the EPR-paradox by David Bohm and Yakir Aharonov [35] showing that the EPR-paradox could be tested experimentally, that there "must be a mechanism whereby the setting of one measuring device can influence the reading of another instrument, however remote." [36] This is known as *quantum entanglement*. From these results we can derive states of equal coefficients and a single measurement from Equation (2.5). These are known as the *Bell states* or *EPR pairs*:

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \quad (2.7)$$

$$|\Phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \quad (2.8)$$

$$|\Psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \quad (2.9)$$

$$|\Psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \quad (2.10)$$

They form the *Bell basis* of the four-dimensional Hilbert space for 2 qubits.

3. QUANTUM CIRCUITS

3.1. QUANTUM GATES.

Classical computers are made of wires and logic gates. Wires move bits around while logic gates manipulate input bits and produce output bits. In order to build a QC, there must be a similar model of computation allowing for input qubits to be converted to output qubits but with one key difference; the QC must be reversible. This is because the laws of quantum mechanics are reversible in time, and as such, any quantum operation must be reversible. These observations were first made by Charles Bennett in 1973 [37] and developed further by Edward Fredkin and Tommaso Toffoli as well as Bennett separately in 1982. [38, 39] Feynman continued this work, discussing possible Hamiltonians for QCs in 1986. [40]

To define a gate classically, we can use a truth table. For example, the truth tables for the NOT and AND gates are:

NOT		AND			(3.1)
a	a'	a	b	a'	
0	1	0	0	0	
0	1	0	1	0	
1	0	1	0	0	
		1	1	1	

If at a minimum we wish QCs to be able to perform the same operations as classical computers, they must be able to perform these 2 operations since they are the minimum required for a Turing complete classical computer (either OR or

AND are required). [41] In order to convert these into quantum gates we must make sure that they are reversible, that is to say, for a quantum mechanical system they must be unitary operators. (An operator U is unitary if $U^\dagger U = U U^\dagger = I$ and $U^\dagger = U^{-1}$ where I is the identity matrix.)

For NOT, we can define the operation as:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |1\rangle + \beta |0\rangle \quad (3.2)$$

where we can write the state as a column vector of complex coefficients:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (3.3)$$

where the top entry is the complex coefficient of $|0\rangle$ and the bottom entry is the complex coefficient of $|1\rangle$. The NOT gate can then be represented as:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.4)$$

If we apply the NOT gate to the state (3.3), we get:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad (3.5)$$

which is also easily shown to be unitary.

AND however is more tricky. Looking at its truth table, we can see that it is not reversible. An output of 0 could be the result of three different inputs. However, in 1981 Toffoli published a paper discussing the construction of reversible primitives to create an arbitrary invertible combinatorial function. [42] The three primitives required are: NOT, CONTROLLED NOT (CNOT) and CONTROLLED CONTROLLED NOT (CCNOT). The truth tables for CNOT and CCNOT are:

CNOT				CCNOT		
a	b	a'	b'	a	b	c
0	0	0	0	0	0	0
0	1	0	1	0	0	1
1	0	1	1	0	1	1
1	1	1	0	1	0	1
				a'	b'	c'
0	0	0	0	0	0	0
0	1	0	1	0	0	1
1	0	1	1	0	1	0
1	1	1	0	1	0	1
				1	1	1
				1	1	0

(3.6)

Already we can see that using CCNOT, by setting $c = 0$, $c' = 1$ only if $a = b = 1$. This is the AND gate. However, crucially the CCNOT gate is reversible due to the fact that it has a unique output for each input.

For CNOT we can define the operation as:

$$\alpha |00\rangle + \beta |01\rangle + \gamma |10\rangle + \delta |11\rangle \rightarrow \alpha |00\rangle + \beta |01\rangle + \delta |10\rangle + \gamma |11\rangle \quad (3.7)$$

The CNOT gate can then be represented as:

$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.8)$$

CCNOT is similar to CNOT, but with an additional control qubit. We can define the operation as:

$$\begin{aligned} & \alpha |000\rangle + \beta |001\rangle + \gamma |010\rangle + \delta |011\rangle + \epsilon |100\rangle + \zeta |101\rangle + \eta |110\rangle + \theta |111\rangle \\ \rightarrow & \alpha |000\rangle + \beta |001\rangle + \gamma |010\rangle + \delta |011\rangle + \epsilon |100\rangle + \zeta |101\rangle + \eta |111\rangle + \theta |110\rangle \end{aligned} \quad (3.9)$$

The CCNOT gate can then be represented as:

$$CCNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.10)$$

While viewing the gates in their matrix forms is useful for tracking the operations each gate makes, a more intuitive way could be to present the gates as circuits more akin to classical computing. This can be seen in Figures 3.1, 3.2 and 3.3.

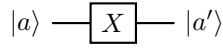


FIGURE 3.1. Circuit for NOT.

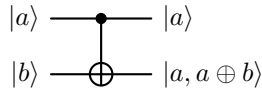


FIGURE 3.2. Circuit for CNOT.

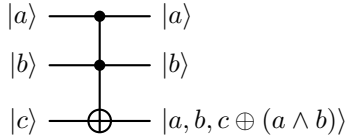


FIGURE 3.3. Circuit for CCNOT.

With these gates, any classical circuit is constructable. For example, the full adder circuit can be constructed using CNOT and CCNOT gates as shown in Figure 3.4.

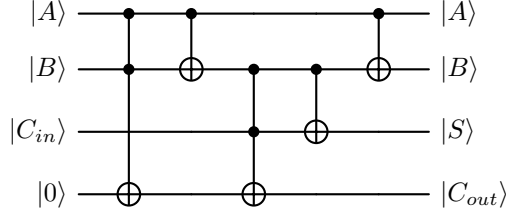


FIGURE 3.4. Circuit for Full Adder where S is the sum and $C_{in/out}$ is the carry in/out.

However, quantum circuits are not limited to classical circuits; any unitary matrix, a theoretically infinite amount, [43] can be used as a quantum logic gate. The most obvious of which, for quantum mechanics would be the Identity and the Pauli matrices [44] which act on a single qubit state, one of which we have already defined in Equation (3.4).

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3.11)$$

$$X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (3.12)$$

$$Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad (3.13)$$

$$Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (3.14)$$

where X is the NOT gate and Y and Z can be visualized as rotations around the Bloch sphere. Y maps $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ to $-i|0\rangle$, while Z maps $|0\rangle$ to $|0\rangle$ and $|1\rangle$ to $-|1\rangle$. Z can be especially useful as it flips the phase of the single qubit state. We could generalize the Z gate to the Phase shift gate:

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \quad (3.15)$$

and redefine the Z gate as $P(\pi)$. However, $P(\varphi)$ is not Hermitian except for if $\varphi = n\pi, n \in \mathbb{Z}$. This means that it will not correspond to a difference in measurement and purely operates as change of relative phase.

We could also generalize the CNOT gate to a CU gate where U is any single qubit gate. This is done by applying the U gate to the target qubit if the control qubit is 1. The CU gate can be represented as:

$$CU = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \quad (3.16)$$

where u_{ij} are the elements of the matrix U .

In Equations (2.2, 2.3) 2 states were introduced $|+\rangle$ and $|-\rangle$, where the state has an equal chance of collapsing to either $|0\rangle$ or $|1\rangle$. If we wanted to create these states from an input state, say $|0\rangle$ or $|1\rangle$, we could use the following gate:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (3.17)$$

which would map $|0\rangle$ to $|+\rangle$ and $|1\rangle$ to $|-\rangle$. This gate is known as the Hadamard gate after Jaques Hadamard. [45] For example, if we apply the Hadamard gate to $|0\rangle$, we get:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle \quad (3.18)$$

Creating a superposition of states is a key part of the power of quantum computing because multiple states can then be simultaneously evaluated. This will be discussed further in Section 4.

Finally, in order to use any information we have computed, we must measure our quantum states. This can be done by measuring (projecting) the state onto a basis. Since this operation is irreversible, it is not a unitary operation and as such, not a quantum gate; implementations of measurement will depend on the physical system used to implement the QC. Using the visual representation of quantum circuits we can represent measurement as shown in Figure 3.5.



FIGURE 3.5. Symbol for measurement of a qubit. The incoming single wire represents that it carries a qubit while the outgoing double wire represents that it carries a classical bit.

3.2. COPYING A QUBIT.

Using a classical CNOT gate, it is elementary to copy a single bit; we take in an unknown bit x and a bit initialised to 0, if x is 0 then the output is 0 and if x is 1 then the output is 1. Suppose we do the same with an unknown qubit $|\psi\rangle$ and a qubit $|0\rangle$:

$$|\psi\rangle|0\rangle = \alpha|00\rangle + \beta|10\rangle \xrightarrow{\text{CNOT}} \alpha|00\rangle + \beta|11\rangle \quad (3.19)$$

Therefore we have changed the state of $|\psi\rangle|0\rangle$ based on the value of $|0\rangle$; we have classically copied information. However, for the general case, where neither qubit is known and the operation must be reversible, a copied qubit would be in the state:

$$|\psi\rangle|\psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle \quad (3.20)$$

where there is an output for every unique input. However, this is not the case for Equation (3.19) unless $\alpha\beta = 0$; cloning a qubit is not possible. This is known as the *no-cloning theorem* and was first proven by Wootters and Zurek in 1982. [46].

Despite not being able to clone a qubit our output from Equation (3.19) is still useful. These are a similar form to the Bell states from Equations (2.7, 2.8, 2.9, 2.10). In fact, we can make a circuit using Hadamard to create a Bell state from 2 qubits. This is shown in Figure 3.6.

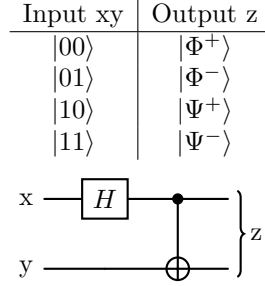


FIGURE 3.6. Circuit for creating a Bell state.

4. QUANTUM ALGORITHMS AND PARALLELISM

4.1. DEUTSCH-JOZSA ALGORITHM.

Originally proposed by David Deutsch and Richard Jozsa in 1992 and updated by Richard Cleve, Artur Ekert, Chiara Macchiavello and Michele Mosca in 1998, [47, 48] the Deutsch-Jozsa algorithm is one of the earliest examples of a quantum algorithm that outperforms its classical counterpart and while it doesn't have much practical use, it nicely shows the parallelism of quantum computing.

The problem that the algorithm aims to solve is as follows: We have a QC (U_f) of which we have no knowledge of its internal workings. It implements some function $f : \{0,1\}^n \rightarrow \{0,1\}$ which takes n -bit binary values as inputs and outputs either a 0 or a 1. Lastly we assume that the function is either constant (always outputs 0 or 1 no matter the input) or balanced (outputs 0 and 1 for half the inputs each). We then wish to determine if the function is constant or balanced.

The classical solution to this problem would be to evaluate the function for all possible inputs and if we find that the function is not constant, then it must be balanced. This would require $2^{n-1} + 1$ evaluations of the function. However, the Deutsch-Jozsa algorithm can solve this problem with only 1 evaluation of the function. The algorithm can be seen in Figure 4.1.

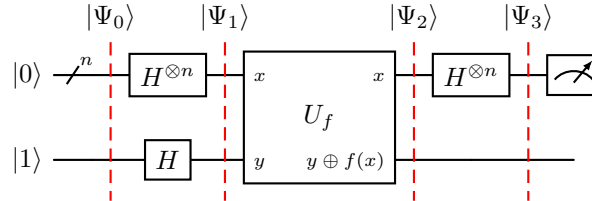


FIGURE 4.1. Circuit for the Deutsch-Jozsa algorithm.

Walking though the algorithm step by step:

- (1) Our initial state $|\Psi_0\rangle$ is with $n + 1$ qubits, n of which are in the state $|0\rangle$ and the last in the state $|1\rangle$. This can also be written as $|0\rangle^{\otimes n} |1\rangle$.

$$|\Psi_0\rangle = |0\rangle^{\otimes n} |1\rangle \quad (4.1)$$

- (2) We then apply the Hadamard gate to all qubits:

$$H |\Psi_0\rangle = |\Psi_1\rangle = \sum_{x=0}^{2^n-1} \frac{|x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (4.2)$$

- (3) We then apply the oracle U_f which acts like $|x\rangle \rightarrow (-1)^{f(x)} |x\rangle$:

$$U_f |\Psi_1\rangle = |\Psi_2\rangle = \sum_x \frac{(-1)^{f(x)} |x\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (4.3)$$

- (4) We then apply the Hadamard gate to the first n qubits:

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)} |z\rangle}{\sqrt{2^n}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \quad (4.4)$$

since for a single qubit $H|x\rangle = \frac{1}{\sqrt{2}} \sum_z (-1)^{x \cdot z} |z\rangle$.

- (5) Finally, we measure $|\Psi_3\rangle$. In order for the function f to be constant, the state must either be $|0\rangle^{\otimes n}$ or $|1\rangle^{\otimes n}$. In these states, the amplitude of the state is $\frac{1}{2^n} \sum_x (-1)^{f(x)}$ which will evaluate to either 1 or -1. Since $|\Psi_3\rangle$ is normalized, this must be the only possible state in the superposition hence we can conclude that the function is constant.

However, if the function is balanced, the amplitude of the state $|0\rangle^{\otimes n}$ or $|1\rangle^{\otimes n}$ will be 0 since the negative and positive contributions will cancel. This means that the state must collapse to a state other than all 0s or all 1s. Since we know the function to be either constant or balanced, we can conclude that the function is balanced.

As a final note on the Deutsch-Jozsa algorithm, the algorithm known as *Deutsch's algorithm* is a specific case of the Deutsch-Jozsa algorithm where $n = 1$. This simplifies the algorithm to only 2 qubits and is an example of a quantum algorithm that performs better than its classical counterpart.

4.2. BERNSTEIN-VAZIRANI ALGORITHM.

The Bernstein-Vazirani algorithm is an extension of the Deutsch-Jozsa algorithm and was first proposed by Ethan Bernstein and Umesh Vazirani in 1997. [49]. The problem is modified so that instead of determining between 2 classes of functions, it instead attempts to find a hidden binary string s of length n encoded in a function. Our new oracle function U_f implements the function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ where $f(x) = x \cdot s$ and $s \in \{0, 1\}^n$.

To compute this classically, we would require n evaluations of the function where we input a string x with only one bit set to 1 in each input, iterating over all possible

strings. This would return each bit of the hidden string s like:

$$\begin{aligned} f(1000 \dots 0) &= s_1 \\ f(0100 \dots 0) &= s_2 \\ &\vdots \\ f(0000 \dots 1) &= s_n \end{aligned}$$

where the bit string x is length n . Then the value of s would be $s = s_1 s_2 \dots s_n$.

However, just like in the Deutsch-Jozsa algorithm, the Bernstein-Vazirani algorithm can be evaluated once to be solved. Figure 4.2 shows the circuit for the algorithm.

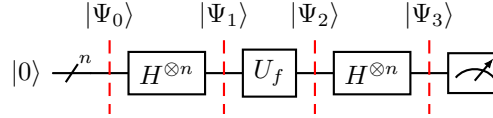


FIGURE 4.2. Circuit for the Bernstein-Vazirani algorithm.

Using the result from Equation (4.4) and removing the qubit that was initialised to $|1\rangle$:

$$|\Psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \cdot z + f(x)} |z\rangle}{\sqrt{2^n}} = |s\rangle \quad (4.5)$$

where we can measure $|s\rangle$ to find the hidden string s .

4.3. SHOR'S ALGORITHM.

As discussed in Section 2.2.1, the security of RSA is dependant on the difficulty of calculating the prime factors of large numbers. Shor's algorithm, proposed by Peter Shor in 1994, is a quantum algorithm that can factorize large numbers in polynomial time. [12]

The RSA algorithm uses 3 large positive integers, e , d and n , where n is the product of 2 large prime numbers, p and q , and for all integers m ($0 \leq m < n$), both $(m^e)^d$ and m have the same remainder when divided by n . That is to say:

$$(m^e)^d \equiv m \pmod{n} \quad (4.6)$$

where n and e make up the public key, d is the private key, and m is the message. We can then define the encryption and decryption as follows:

$$c \equiv m^e \pmod{n} \quad (4.7)$$

$$m \equiv c^d \pmod{n} \quad (4.8)$$

where c is the cipher text.

The RSA algorithm is said to be secure because breaking it requires recovering m such that Equation (4.7) is true. To do this requires the factorization of n into its prime factors, hence allowing the calculation of d from e and the prime factors of n .

The first observation to make is that, if n is even, then 2 is a factor. This can be done easily using the Euclidean algorithm. [50] However, if n is odd, and not a prime power (which we can check efficiently using an algorithm by Bernstein [51]) then we can use Shor's algorithm to factorize n .

Shor's algorithm is as follows:

- (1) Choose a random number a where $1 < a < n$.
- (2) Find the greatest common divisor K of a and n ; $K = \gcd(a, n)$.
- (3) If $K \neq 1$, then K must be a non-trivial factor and our second factor is $\frac{n}{K}$.
- (4) If $K = 1$, then n and a are coprime, so a has a multiplicative order (smallest positive integer) r modulo n such that $a^r \equiv 1 \pmod{n}$.
- (5) To find r we can use the quantum subroutine in Figure 4.3.
- (6) If r is odd, retry with a different a from step 1.
- (7) If r is even, then we can find the factors of n as $p = \gcd(a^{r/2} + 1, n)$ and (if p is non-trivial) $q = \frac{n}{p}$ where p and q are the prime factors of n .

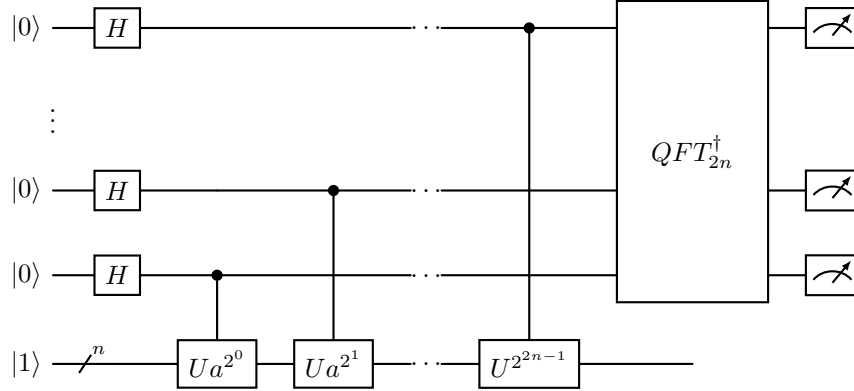


FIGURE 4.3. Quantum subroutine for Shor's algorithm from Figure 1 in Circuit for Shor's algorithm using $2n + 3$ qubits by Stephane Beauregard. [52]

Figure 4.3 shows the quantum subroutine for Shor's algorithm using $2n + 3$ qubits. It uses *quantum phase estimation* [53] by a *quantum Fourier transform* [B] on an input state $|0\rangle^{\otimes n} |1\rangle$:

$$|0\rangle^{\otimes n} |1\rangle \rightarrow \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |\phi_j\rangle |\psi_j\rangle \quad (4.9)$$

where $|\phi_j\rangle$ is the binary representation of j and $|\psi_j\rangle$ is an eigenvector of U with eigenvalue $e^{2\pi i j \theta}$. If we measure the first register, we have a $1/r$ chance of measuring each $|\phi_j\rangle$. Measuring each $|\phi_j\rangle$ gives an integer approximation $2^{2n} j/r$.

Now we have p and q , we can trivially calculate d as $d = e^{-1} \pmod{\lambda(n)}$ where $\lambda(n) = \text{lcm}(p-1, q-1)$.

5. EXPERIMENTAL QUANTUM COMPUTING

What are the experimental challenges in building a QC? In 1996 David DiVincenzo proposed a list of criteria that a physical system that can be used to implement a QC must meet to be considered useful. [54]

The system:

- (1) is scalable and qubits are well characterized.
- (2) can be placed into a fiducial starting quantum state.
- (3) is isolated from the environment so that decoherence times are long.
- (4) must be able to be subjected to a controlled sequence of unitary transformations.
- (5) must be able to be project a state irreversibly into its eigenfunction, determining which orthogonal eigenstate the state is in; the qubit is measurable.

Out of these requirements, Divincenzo suggests that the hardest is (3) while maintaining (4) and (5). For example, NMR QCs have relatively long decoherence times [55] but since they are isolated, they are insufficiently sensitive to detect the states of individual spins resulting in "weak" measurements where the state is not fully projected. [56]

There have been many different systems that have been used to implement a QC, each with their own advantages and disadvantages but all must meet the criteria set out by DiVincenzo if they are to provide a useful QC. For an overview of the different systems, see the Figure A.1.

6. CONCLUSION

Quantum computing is a growing rapidly thanks to the ability for quantum computers to solve problems impossible for classical computers. Recent advances see larger numbers of qubits finally enabling algorithms such as Shor's algorithm to be used in practice, allowing quantum algorithms, with their inherent parallelism, to outperform classical algorithms. Using the basic building blocks in this report, many more useful applications of quantum computing can be derived; solving optimization problems, simulating quantum systems, data analysis, cybersecurity and more can benefit.

APPENDIX A.
ADDITIONAL INFORMATION

System	Qubits	Initial State	Unitary Transformations	Measurement
Optical Photon [57]	Polarization of photons.	Single photon states (e.g. attenuation of lasers).	Phase shifters, beam splitters. Couplings via non-linear Kerr media.	Photo-detectors.
Ion trap [58]	Nuclear spin of atom and phonons of trapped atoms.	Cool the atoms into ground state.	Laser pulses externally manipulate the atomic state via the Jaynes-Cummings interaction. Qubits couple with phonon state.	Measure population of hyperfine states.
NMR [59]	Spin of atomic nucleus.	Polarize spins with strong magnetic field.	Magnetic field pulses manipulate the spins. Couplings are chemical bonds.	Voltage induced by precession of spins.
Superconducting qubits [60]	Charge, flux or phase of superconducting circuits.	Wait for thermalization.	Pulsed microwave signals. Couplings are capacitive.	Electrometer or magnetometer if using flux qubits.

FIGURE A.1. Experimental implementations of quantum computing.

APPENDIX B.
QUANTUM FOURIER TRANSFORM

The quantum Fourier transform (QFT) is same as the classical discrete Fourier transform but applied to the vector of amplitudes of a quantum state. On a register of n qubits, the vector of amplitudes is of length $N = 2^n$.

The classical Fourier transform is defined as:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{-jk} \quad (k=0, 1, \dots, N-1)$$

where $\omega_N = e^{2\pi i/N}$ and $x : (x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$ and $y : (y_0, y_1, \dots, y_{N-1}) \in \mathbb{C}^N$ are the input and output vectors respectively. Since the classical Fourier transform

already acts on a vector, no change is needed to apply it to a quantum state, mapping:

$$|x\rangle = \sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} y_k |k\rangle \quad (\text{B.1})$$

where:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j \omega_N^{jk} \quad (k=0, 1, \dots, N-1)$$

and ω_N^{jk} is the phase factor. For the QFT the convention is a positive sign.

In the case where $|x\rangle$ is a basis state, the QFT can be expressed as a map of $|x\rangle \rightarrow |k\rangle$:

$$\text{QFT } |x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle \quad (\text{B.2})$$

which we can write as a unitary matrix:

$$\text{QFT} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \omega_N & \omega_N^2 & \omega_N^3 & \dots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \omega_N^6 & \dots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \omega_N^9 & \dots & \omega_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \omega_N^{3(N-1)} & \dots & \omega_N^{(N-1)^2} \end{bmatrix} \quad (\text{B.3})$$

To implement the QFT, we can use the circuit shown in Figure B.1.

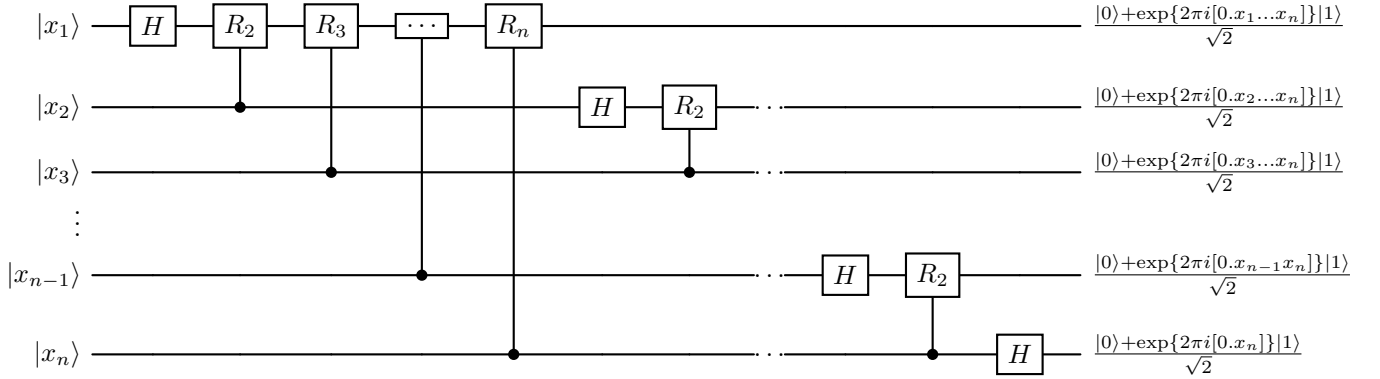


FIGURE B.1. Circuit for the Quantum Fourier Transform where R_k is the controlled rotation gate $R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$ and the notation $0.x$ is the binary fraction like $0.x_1x_2 \dots x_n = \frac{x_1}{2^1} + \frac{x_2}{2^2} + \dots + \frac{x_n}{2^n}$.

APPENDIX C. FURTHER READING

Included here are some further reading suggestions.

- On the differences of problem class and the increase of theoretic power over the probabilistic Turing Machine [61]
- On the Travelling Salesman Problem [62]
- On the role of light based approaches in quantum computing [63]
- On the reduction of errors and inaccuracies [64]
- On the improvement of search algorithms by quantum walk [65]
- On the simulation of the evolution of sparse Hamiltonians [66]
- On the introduction of surface code quantum computing [67]
- On the use of simple single-qubit rotations for Hamiltonian simulation [68]
- On the Solovay-Kitaev theorem in order to compile quantum circuits into efficient, fault tolerant circuits [69]

REFERENCES

- [1] WEISS, E.: Konrad Zuse Obituary. In: *IEEE Annals of the History of Computing* 18 (1996), Nr. 2, S. 3–. <http://dx.doi.org/10.1109/MAHC.1996.489747>. – DOI 10.1109/MAHC.1996.489747
- [2] MOORE, Gordon E.: Cramming more components onto integrated circuits, Reprinted from *Electronics*, volume 38, number 8, April 19, 1965, pp.114 ff. In: *IEEE Solid-State Circuits Society Newsletter* 11 (2006), Nr. 3, S. 33–35. <http://dx.doi.org/10.1109/N-SSC.2006.4785860>. – DOI 10.1109/N-SSC.2006.4785860
- [3] SAMSUNG: *Samsung begins chip production using 3NM Process Technology with GAA Architecture.* <https://news.samsung.com/global/samsung-begins-chip-production-using-3nm-process-technology-with-gaa-architecture>. Version: Jun 2022
- [4] YANG, N. ; HENSON, W.K. ; WORTMAN, J.J.: A comparative study of gate direct tunneling and drain leakage currents in n-MOSFET's with sub-2 nm gate oxides. In: *IEEE Transactions on Electron Devices* 47 (2000), Nr. 8, S. 1636–1644. <http://dx.doi.org/10.1109/16.853042>. – DOI 10.1109/16.853042
- [5] BENIOFF, P.: The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. In: *Journal of Statistical Physics* 22 (1980), Nr. 5, S. 563–591. <http://dx.doi.org/10.1007/BF01011339>. – DOI 10.1007/BF01011339
- [6] MANIN, Yu. I.: Vychislimoe i nevychislimoe. In: *Sovetskoe Radio* (1980), S. 13–15
- [7] FEYNMAN, R. P.: Simulating physics with computers. In: *International Journal of Theoretical Physics* 21 (1982), Nr. 6/7, 467–488. <http://dx.doi.org/10.1007/BF02650179>. – DOI 10.1007/BF02650179
- [8] BENNETT, Charles H. ; BRASSARD, Gilles: Quantum cryptography: Public key distribution and coin tossing. In: *Theoretical Computer Science* 560 (2014), Dezember, 7–11. <http://dx.doi.org/10.1016/j.tcs.2014.05.025>. – DOI 10.1016/j.tcs.2014.05.025. – ISSN 0304–3975
- [9] DEUTSCH, David: Quantum theory, the Church-Turing principle and the universal quantum computer. In: *Proceedings of the Royal Society of London A* 400 (1985), Nr. 1818, 97–117. <http://dx.doi.org/10.1098/rspa.1985.0070>. – DOI 10.1098/rspa.1985.0070
- [10] BERNSTEIN, Ethan ; VAZIRANI, Umesh: Quantum complexity theory. (1993), 11–20. <http://dx.doi.org/10.1145/167088.167097>. – DOI 10.1145/167088.167097. ISBN 0897915917
- [11] SIMON, D.R.: On the power of quantum computation. (1994), 116–123. <http://dx.doi.org/10.1109/SFCS.1994.365701>. – DOI 10.1109/SFCS.1994.365701
- [12] SHOR, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. (1994), 124–134. <http://dx.doi.org/10.1109/SFCS.1994.365700>. – DOI 10.1109/SFCS.1994.365700
- [13] GROVER, Lov K.: A fast quantum mechanical algorithm for database search. (1996). <http://dx.doi.org/10.48550/arXiv.quant-ph/9605043>. – DOI 10.48550/arXiv.quant-ph/9605043

- [14] LLOYD, Seth: Universal Quantum Simulators. In: *Science* 273 (1996), Nr. 5278, 1073-1078. <http://dx.doi.org/10.1126/science.273.5278.1073>. – DOI 10.1126/science.273.5278.1073
- [15] CHUANG, Isaac L. ; GERSHENFELD, Neil ; KUBINEC, Mark: Experimental Implementation of Fast Quantum Searching. In: *Phys. Rev. Lett.* 80 (1998), Apr, 3408–3411. <http://dx.doi.org/10.1103/PhysRevLett.80.3408>. – DOI 10.1103/PhysRevLett.80.3408
- [16] VANDERSYPEN, Lieven M. K. ; STEFFEN, Matthias ; BREYTA, Gregory ; YANNONI, Costantino S. ; SHERWOOD, Mark H. ; CHUANG, Isaac L.: Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance. In: *Nature* 414 (2001), Dezember, Nr. 6866, 883–887. <http://dx.doi.org/10.1038/414883a>. – DOI 10.1038/414883a. – ISSN 1476–4687
- [17] NAKAMURA, Y. ; PASHKIN, Yu. A. ; TSAI, J. S.: Coherent control of macroscopic quantum states in a single-Cooper-pair box. In: *Nature* 398 (1999), Apr, 786-788. <http://dx.doi.org/10.1038/19718>. – DOI 10.1038/19718
- [18] PLANTENBERG, J. H. ; GROOT, P. C. ; HARMANS, C. J. P. M. ; MOOLJ, J. E.: Demonstration of controlled-NOT quantum gates on a pair of superconducting quantum bits. In: *Nature* 447 (2007), Jun, 836-839. <http://dx.doi.org/10.1038/nature05896>. – DOI 10.1038/nature05896
- [19] ATOMCOMPUTING: *Quantum startup Atom Computing first to exceed 1,000 qubits*. <https://atom-computing.com/quantum-startup-atom-computing-first-to-exceed-1000-qubits/>. Version: Oct 2023
- [20] NORCIA, M. A. ; KIM, H. ; CAIRNCROSS, W. B. ; STONE, M. ; RYOU, A. ; JAFFE, M. ; BROWN, M. O. ; BARNES, K. ; BATTAGLINO, P. ; BOHDANOWICZ, T. C. ; BROWN, A. ; CASSELLA, K. ; CHEN, C.-A. ; COXE, R. ; CROW, D. ; EPSTEIN, J. ; GRIGER, C. ; HALPERIN, E. ; HUMMEL, F. ; JONES, A. M. W. ; KINDEM, J. M. ; KING, J. ; KOTRU, K. ; LAUGAN, J. ; LI, M. ; LU, M. ; MEGIDISH, E. ; MARJANOVIC, J. ; McDONALD, M. ; MITTIGA, T. ; MUNIZ, J. A. ; NARAYANASWAMI, S. ; NISHIGUCHI, C. ; PAULE, T. ; PAWLAK, K. A. ; PENG, L. S. ; PUDENZ, K. L. ; RODRÍGUEZ PÉREZ, D. ; SMULL, A. ; STACK, D. ; URBANEK, M. ; VEERDONK, R. J. M. d. ; VENDEIRO, Z. ; WADLEIGH, L. ; WILKASON, T. ; WU, T.-Y. ; XIE, X. ; ZALYS-GELLER, E. ; ZHANG, X. ; BLOOM, B. J.: Iterative Assembly of ^{171}Yb Atom Arrays with Cavity-Enhanced Optical Lattices. In: *PRX Quantum* 5 (2024), Jul, 030316. <http://dx.doi.org/10.1103/PRXQuantum.5.030316>. – DOI 10.1103/PRXQuantum.5.030316
- [21] IBM: *The hardware and software for the era of quantum utility is here*. <https://www.ibm.com/quantum/blog/quantum-roadmap-2033>. Version: Dec 2023
- [22] ABUGHANEM, M.: *IBM Quantum Computers: Evolution, Performance, and Future Directions*. <https://arxiv.org/abs/2410.00916>. Version: 2024
- [23] DIFFIE, W. ; HELLMAN, M.: New directions in cryptography. In: *IEEE Transactions on Information Theory* 22 (1976), Nr. 6, S. 644–654. <http://dx.doi.org/10.1109/TIT.1976.1055638>. – DOI 10.1109/TIT.1976.1055638
- [24] RIVEST, R. L. ; SHAMIR, A. ; ADLEMAN, L.: A method for obtaining digital signatures and public-key cryptosystems. In: *Commun. ACM* 21 (1978), Februar, Nr. 2, 120–126. <http://dx.doi.org/10.1145/359340.359342>. – DOI 10.1145/359340.359342. – ISSN 0001–0782
- [25] BRIGGS, M. E.: An Introduction to the General Number Field Sieve. (1998). https://personal.math.vt.edu/brown/doc/briggs_gnfs_thesis.pdf
- [26] BECKMAN, David ; CHARI, Amalavoyal N. ; DEVABHAKTUNI, Srikrishna ; PRESKILL, John: Efficient networks for quantum factoring. In: *Phys. Rev. A* 54 (1996), Aug, 1034–1063. <http://dx.doi.org/10.1103/PhysRevA.54.1034>. – DOI 10.1103/PhysRevA.54.1034
- [27] HARVEY, David ; HOEVEN, Joris van d.: Integer multiplication in time $O(n \log n)$. In: *Annals of Mathematics* 193 (2021), Nr. 2, 563 – 617. <http://dx.doi.org/10.4007/annals.2021.193.2.4>. – DOI 10.4007/annals.2021.193.2.4
- [28] SCARFONE, K. ; JANSEN, W. ; TRACY, M.: Guide to General Server Security. In: *NIST Special Publication* 800-123 (2008). <http://dx.doi.org/10.6028/NIST.SP.800-123>. – DOI 10.6028/NIST.SP.800-123
- [29] BENNETT, Charles H. ; BERNSTEIN, Ethan ; BRASSARD, Gilles ; VAZIRANI, Umesh: Strengths and Weaknesses of Quantum Computing. In: *SIAM Journal on Computing* 26 (1997), Nr. 5, 1510-1523. <http://dx.doi.org/10.1137/S0097539796300933>. – DOI 10.1137/S0097539796300933
- [30] SCHRÖDINGER, E.: An Undulatory Theory of the Mechanics of Atoms and Molecules. In: *Phys. Rev.* 28 (1926), Dec, 1049–1070. <http://dx.doi.org/10.1103/PhysRev.28.1049>. – DOI 10.1103/PhysRev.28.1049

- [31] HUBBARD, J.: Electron Correlations in Narrow Energy Bands. In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 276 (1963), Nov, Nr. 1365, 238–257. <http://dx.doi.org/10.1098/rspa.1963.0204>. – DOI 10.1098/rspa.1963.0204
- [32] AARONSON, Scott: *Quantum Computing since Democritus*. Cambridge University Press, 2013. – 132 S. <https://doi.org/10.1017/CB09780511976667>
- [33] FEYNMAN, Richard P. ; VERNON, Jr. Frank L. Frank L. ; HELLWARTH, Robert W.: Geometrical Representation of the Schrödinger Equation for Solving Maser Problems. In: *Journal of Applied Physics* 28 (1957), 01, Nr. 1, 49–52. <http://dx.doi.org/10.1063/1.1722572>. – DOI 10.1063/1.1722572. – ISSN 0021–8979
- [34] EINSTEIN, A. ; PODOLSKY, B. ; ROSEN, N.: Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? In: *Phys. Rev.* 47 (1935), May, 777–780. <http://dx.doi.org/10.1103/PhysRev.47.777>. – DOI 10.1103/PhysRev.47.777
- [35] BOHM, D. ; AHARONOV, Y.: Discussion of Experimental Proof for the Paradox of Einstein, Rosen, and Podolsky. In: *Phys. Rev.* 108 (1957), Nov, 1070–1076. <http://dx.doi.org/10.1103/PhysRev.108.1070>. – DOI 10.1103/PhysRev.108.1070
- [36] BELL, J. S.: On the Einstein Podolsky Rosen paradox. In: *Physics Physique Fizika* 1 (1964), Nov, 195–200. <http://dx.doi.org/10.1103/PhysicsPhysiqueFizika.1.195>. – DOI 10.1103/PhysicsPhysiqueFizika.1.195
- [37] BENNETT, C. H.: Logical Reversibility of Computation. In: *IBM Journal of Research and Development* 17 (1973), Nr. 6, S. 525–532. <http://dx.doi.org/10.1147/rd.176.0525>. – DOI 10.1147/rd.176.0525
- [38] FREDKIN, E. ; TOFFOLI, T.: Conservative Logic. In: *International Journal of Theoretical Physics* 21 (1982), Nr. 3/4, 219–253. <http://dx.doi.org/10.1007/BF01857727>. – DOI 10.1007/BF01857727
- [39] BENNETT, C. H.: The thermodynamics of computation—a review. In: *International Journal of Theoretical Physics* 21 (1982), Dec, Nr. 12, 905–940. <http://dx.doi.org/10.1007/BF02100230>. – DOI 10.1007/BF02100230
- [40] FEYNMAN, R. P.: Quantum mechanical computers. In: *Foundations of Physics* 16 (1986), Nr. 6, 507–531. <http://dx.doi.org/10.1007/BF01886518>. – DOI 10.1007/BF01886518
- [41] COPI, C. J. ; COHEN, M. ; MCMAHON, S.: Introduction to Logic. (2011). <http://dx.doi.org/10.4324/9781315510897>. – DOI 10.4324/9781315510897. ISBN 9781315510897
- [42] TOFFOLI, T.: Bicontinuous extensions of invertible combinatorial functions. In: *Mathematical Systems Theory* 14 (1981), 13–23. <http://dx.doi.org/10.1007/BF01752388>. – DOI 10.1007/BF01752388
- [43] BARENCO, Adriano ; BENNETT, Charles H. ; CLEVE, Richard ; DiVINCENZO, David P. ; MARGOLUS, Norman ; SHOR, Peter ; SLEATOR, Tycho ; SMOLIN, John A. ; WEINFURTER, Harald: Elementary gates for quantum computation. In: *Phys. Rev. A* 52 (1995), Nov, 3457–3467. <http://dx.doi.org/10.1103/PhysRevA.52.3457>. – DOI 10.1103/PhysRevA.52.3457
- [44] PAULI, W.: Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren. In: *Zeitschrift für Physik* 31 (1925), Feb, Nr. 1, 765–783. <http://dx.doi.org/10.1007/BF02980631>. – DOI 10.1007/BF02980631
- [45] HADAMARD, J.S.: Résolution d’une question relative aux déterminants. In: *Bulletin des Sciences Mathématiques* 17 (1893), S. 240–246
- [46] WOOTTERS, W. K. ; ZUREK, W. H.: A single quantum cannot be cloned. In: *Nature* 299 (1982), Oct, 802–803. <http://dx.doi.org/10.1038/299802a0>. – DOI 10.1038/299802a0
- [47] DEUTSCH, David ; JOZSA, Richard: Rapid solution of problems by quantum computation. In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439 (1992), Dec, Nr. 1907, 553–558. <http://dx.doi.org/10.1098/rspa.1992.0167>. – DOI 10.1098/rspa.1992.0167
- [48] CLEVE, Richard ; EKERT, Artur ; MACCHIAVELLO, Chiara ; MOSCA, Michele: Quantum algorithms revisited. In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 454 (1998), Jan, 339–354. <http://dx.doi.org/10.1098/rspa.1998.0164>. – DOI 10.1098/rspa.1998.0164
- [49] BERNSTEIN, Ethan ; VAZIRANI, Umesh: Quantum Complexity Theory. In: *SIAM Journal on Computing* 26 (1997), Nr. 5, 1411–1473. <http://dx.doi.org/10.1137/S0097539796300921>. – DOI 10.1137/S0097539796300921

- [50] SHALLIT, J.: Origins of the analysis of the Euclidean algorithm. In: *Historia Mathematica* 21 (1994), Nr. 4, 401-419. <http://dx.doi.org/https://doi.org/10.1006/hmat.1994.1031>. – DOI <https://doi.org/10.1006/hmat.1994.1031>. – ISSN 0315-0860
- [51] BERNSTEIN, D. J.: Detecting perfect powers in essentially linear time. In: *Mathematics of Computation* 67 (1998), Nr. 224, 1253-1283. <http://dx.doi.org/10.1090/S0025-5718-98-00952-1>. – DOI 10.1090/S0025-5718-98-00952-1
- [52] BEAUREGARD, Stephane: Circuit for Shor's algorithm using $2n+3$ qubits. (2003). <https://arxiv.org/abs/quant-ph/0205095>
- [53] KITAEV, A. Y.: Quantum measurements and the Abelian Stabilizer Problem. (1995). <https://arxiv.org/abs/quant-ph/9511026>
- [54] DiVINCENZO, David P.: Topics in Quantum Computers. (1996). <https://arxiv.org/abs/cond-mat/9612126>
- [55] BLOEMBERGEN, N. ; PURCELL, E. M. ; POUND, R. V.: Relaxation Effects in Nuclear Magnetic Resonance Absorption. In: *Phys. Rev.* 73 (1948), Apr, 679-712. <http://dx.doi.org/10.1103/PhysRev.73.679>. – DOI 10.1103/PhysRev.73.679
- [56] CORY, David G. ; FAHMY, Amr F. ; HAVEL, Timothy F.: Ensemble Quantum Computing by NMR Spectroscopy. In: *Proceedings of the National Academy of Sciences* 94 (1997), Nr. 5, 1634-1639. <http://dx.doi.org/10.1073/pnas.94.5.1634>. – DOI 10.1073/pnas.94.5.1634
- [57] KNILL, E. ; LAFLAMME, R. ; MILBURN, G. J.: A scheme for efficient quantum computation with linear optics. In: *Nature* 409 (2001), 46-52. <http://dx.doi.org/10.1038/35051009>. – DOI 10.1038/35051009
- [58] CIRAC, J. I. ; ZOLLER, P.: Quantum Computations with Cold Trapped Ions. In: *Phys. Rev. Lett.* 74 (1995), May, 4091-4094. <http://dx.doi.org/10.1103/PhysRevLett.74.4091>. – DOI 10.1103/PhysRevLett.74.4091
- [59] LLOYD, Seth: A Potentially Realizable Quantum Computer. In: *Science* 261 (1993), Nr. 5128, 1569-1571. <http://dx.doi.org/10.1126/science.261.5128.1569>. – DOI 10.1126/science.261.5128.1569
- [60] HUANG, He-Liang ; WU, Dachao ; FAN, Daojin ; ZHU, Xiaobo: Superconducting quantum computing: a review. In: *Science China Information Sciences* 63 (2020), jul, Nr. 8. <http://dx.doi.org/10.1007/s11432-020-2881-9>. – DOI 10.1007/s11432-020-2881-9. – ISSN 1869-1919
- [61] SIMON, Daniel R.: On the Power of Quantum Computation. In: *SIAM Journal on Computing* 26 (1997), Nr. 5, 1474-1483. <http://dx.doi.org/10.1137/S0097539796298637>. – DOI 10.1137/S0097539796298637
- [62] GOLDSMITH, Daniel ; DAY-EVANS, Joe: Beyond QUBO and HOBQ formulations, solving the Travelling Salesman Problem on a quantum boson sampler. (2024). <https://arxiv.org/abs/2406.14252>
- [63] WALMSLEY, Ian: Light in quantum computing and simulation: perspective. In: *Optica Quantum* 1 (2023), Oct, Nr. 1, 35-40. <http://dx.doi.org/10.1364/OPTICAQ.507527>. – DOI 10.1364/OPTICAQ.507527
- [64] AHARONOV, Dorit ; BEN-OR, Michael: Fault-Tolerant Quantum Computation With Constant Error Rate. (1999). <https://arxiv.org/abs/quant-ph/9906129>
- [65] MAGNIEZ, Frédéric ; NAYAK, Ashwin ; ROLAND, Jérémie ; SANTHA, Miklos: Search via Quantum Walk. In: *SIAM Journal on Computing* 40 (2011), jan, Nr. 1, 142-164. <http://dx.doi.org/10.1137/090745854>. – DOI 10.1137/090745854. – ISSN 1095-7111
- [66] BERRY, Dominic W. ; AHOKAS, Graeme ; CLEVE, Richard ; SANDERS, Barry C.: Efficient Quantum Algorithms for Simulating Sparse Hamiltonians. In: *Communications in Mathematical Physics* 270 (2006), Dezember, Nr. 2, 359-371. <http://dx.doi.org/10.1007/s00220-006-0150-x>. – DOI 10.1007/s00220-006-0150-x. – ISSN 1432-0916
- [67] FOWLER, Austin G. ; MARIANTONI, Matteo ; MARTINIS, John M. ; CLELAND, Andrew N.: Surface codes: Towards practical large-scale quantum computation. In: *Physical Review A* 86 (2012), September, Nr. 3. <http://dx.doi.org/10.1103/physreva.86.032324>. – DOI 10.1103/physreva.86.032324. – ISSN 1094-1622
- [68] LOW, Guang H. ; CHUANG, Isaac L.: Optimal Hamiltonian Simulation by Quantum Signal Processing. In: *Physical Review Letters* 118 (2017), Januar, Nr. 1. <http://dx.doi.org/10.1103/physrevlett.118.010501>. – DOI 10.1103/physrevlett.118.010501. – ISSN 1079-7114
- [69] DAWSON, Christopher M. ; NIELSEN, Michael A.: The Solovay-Kitaev algorithm. (2005). <https://arxiv.org/abs/quant-ph/0505030>