QUANTUM COMPUTING

ELLIOTT ASHBY
PHYSICS AND ASTRONOMY
UNIVERSITY OF SOUTHAMPTON

Abstract. Placeholder for abstract.

Contents

Date: December 2024.

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 the transistor 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 Brief 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 application of 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]
Just 2 years following this, 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] Just 2 years later in 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 can be programmed to simulate any local quantum system. [14]

The first quantum computer to be built was in 1998; Isaac Chuang and Neil Gershenfeld along with Mark Kubinec implemented Grover's search algorithm with a 2-qubit (the quantum equivalent to bits) nuclear magnetic resonance (NMR) quantum computer. [15] In the years following, NMR quantum computers would increase in the number of qubits allowing for a 7-qubit quantum computer 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 quantum computer with the most number of qubits is actually not a superconducting quantum computer, but an atomic array quantum computer built by Atom Computing in 2023 [19, 20] with a reported 1,180 qubits. However, since atomic array quantum computers are much newer than superconducting quantum computers, they have less general support for quantum algorithms; the largest superconducting quantum computer as of 2024 is IBM's Condor with 1121 qubits. [21, 22]

The field of quantum computing is still in it's infancy, but with promising results and continued growth, it is likely that quantum computers will become more relevant in the coming years.

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 quantum computer 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 up over the GNFS and implies that the RSA algorithm is no longer secure. Among security 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 2.6.4.

2.2.2. *Brute-force Search.*

Brute-force search is a method of problem solving that involves systematically generating and testing all possible 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 significant speed up 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 is said to be asymptotically optimal if it, for large inputs, performs at worst a constant factor worse than the best possible algorithm. Grover's Algorithm is discussed further in section 2.6.5.

2.2.3. *Simulation of Quantum Systems.*

REDO—

Can physics be simulated on a classical computer? In 1982, Richard Feynman states that it is certainly not possible 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, it's 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 $\Psi(x_1, x_2, \ldots, x_n, t)$. 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 it's 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.

This leaves only one option, to simulate quantum systems on a quantum computer. 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 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 qunatum bit or qubit is the equivalent for quantum information theory. [31] 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:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

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, with a probability of $|\alpha|^2$, or 1, with a probability of $|\beta|^2$. An

example of this could be for a qubit to be in the following states:

$$|+\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{2.2}$$

$$|-\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \tag{2.3}$$

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

If we wish to, we can also represent a single qubit using a *Bloch sphere* [32]; a sphere with radius 1 and the poles representing the 2 states $|0\rangle$ and $|1\rangle$. We can rewrite Equation (2.1) as:

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

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

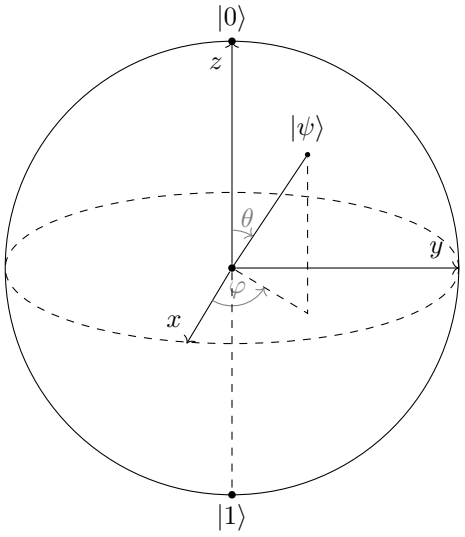

FIGURE 2.1. The Bloch Sphere

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 an infinite amount of data. This is because in order to extract any 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.

Just as 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:

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \tag{2.5}$$

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, then new state of the system would then be:

$$|\psi'\rangle = \frac{\alpha_{10}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}} \tag{2.6}$$

But this raises a strange point, as pointed out by Albert Einstein, Boris Podolsky and Nathan Rosen in 1935 [33] 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 [34] 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." [35] 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*:

$$\left|\Phi^+\right\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.7}$$

$$\left|\Phi^-\right\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \tag{2.8}$$

$$\left|\Psi^+\right\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \tag{2.9}$$

$$\left|\Psi^-\right\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \tag{2.10}$$

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

2.5. Quantum Circuits.

2.5.1. *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 quantum computer, there must be a similar model of computation allowing for input qubits to be converted to output qubits but with one key difference; the quantum computer 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 [36] and developed further by Edward Fredkin and Tommaso Toffoli as well as Bennett separately in 1982. [37, 38] Feynman continues this work, discussing

possible Hamiltonions for quantum computers in 1986. [39]

To define a gate classically, we can use a truth table. For example, here are some classical gates and their truth tables:

NOT	
a	a'
0	1
1	0

AND		
a	b	a'
0	0	0
0	1	0
1	0	0
1	1	1

(2.11)

If at a minimum we wish quantum computers 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). [40] 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 $U^\dagger U = UU^\dagger = I$ and $U^\dagger = U^{-1}$ where I is the identity matrix.)

For NOT, we can define the operation as:

$$\alpha |0\rangle + \beta |1\rangle \rightarrow \alpha |1\rangle + \beta |0\rangle \tag{2.12}$$

if we write the quantum state as:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{2.13}$$

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:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.14}$$

If we apply the NOT gate to the state (2.13), we get:

$$X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \tag{2.15}$$

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. [41] The three primitives required are: NOT, CONTROLLED NOT (CNOT) and CONTROLLED

CONTROLLED NOT (CCNOT). The truth tables for CNOT and CCNOT are:

CNOT

a	b	a'	b'
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

CCNOT

a	b	c	a'	b'	c'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

(2.16)

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:

$$\alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle \rightarrow \alpha\,|00\rangle + \beta\,|01\rangle + \delta\,|10\rangle + \gamma\,|11\rangle \tag{2.17}$$

if we write the quantum state as:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \end{bmatrix} \tag{2.18}$$

where the entries are the complex coefficients of $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$ respectively. The CNOT gate can then be represented as:

$$CNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.19}$$

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

$$\begin{aligned} &\alpha\,|000\rangle + \beta\,|001\rangle + \gamma\,|010\rangle + \delta\,|011\rangle + \epsilon\,|100\rangle + \zeta\,|101\rangle + \eta\,|110\rangle + \theta\,|111\rangle \\ \rightarrow &\alpha\,|000\rangle + \beta\,|001\rangle + \gamma\,|010\rangle + \delta\,|011\rangle + \epsilon\,|100\rangle + \zeta\,|101\rangle + \eta\,|111\rangle + \theta\,|110\rangle \end{aligned} \tag{2.20}$$

If we write the quantum state as:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \\ \delta \\ \epsilon \\ \zeta \\ \eta \\ \theta \end{bmatrix} \tag{2.21}$$

where the entries are the complex coefficients of $|000\rangle$, $|001\rangle$, $|010\rangle$, $|011\rangle$, $|100\rangle$, $|101\rangle$, $|110\rangle$ and $|111\rangle$ respectively. The CCNOT gate can then be represented as:

$$CCNOT \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.22}$$

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 2.2, 2.3 and 2.4.

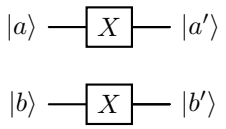

FIGURE 2.2. Circuit for NOT

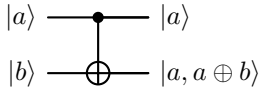

FIGURE 2.3. Circuit for CNOT

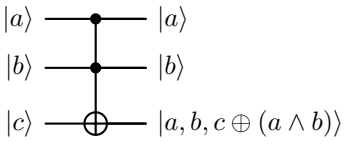

FIGURE 2.4. Circuit for CCNOT

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 2.5.

However, quantum circuits are not limited to classical circuits; any unitary matrix, a theoretically infinite amount, [42] can be used as a quantum logic gate. The most obvious of which, for quantum mechanics would be the Identity and the

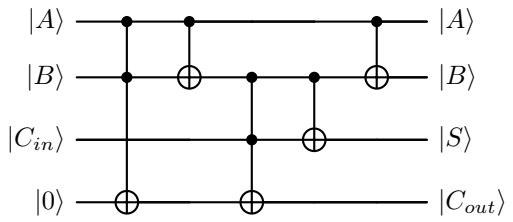

FIGURE 2.5. Circuit for Full Adder where S is the sum and $C_{in/out}$ is the carry in/out

Pauli matrices [43] which act on a single qubit state, one of which we have already defined in Equation (2.14).

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.23}$$

$$X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.24}$$

$$Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.25}$$

$$Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.26}$$

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:

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{2.27}$$

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 coresond 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:

$$CU = \begin{bmatrix} I & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix} \tag{2.28}$$

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:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.29}$$

which would map $|0\rangle$ to $|+\rangle$ and $|1\rangle$ to $|-\rangle$. This gate is known as the Hadamard gate after Jaques Hadamard. [44] For example, if we apply the Hadamard gate to $|0\rangle$, we get:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = |+\rangle \tag{2.30}$$

Creating a superposition of states is a key part of the power of quantum computing, as it allows for the simultaneous evaluation of multiple states. This will be discussed further in Section 2.6.

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 irreversable, 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 quantum computer. Using the visual representation of quantum circuits we can represent measurement as shown in Figure 2.6.

FIGURE 2.6. 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.

2.5.2. *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 for a qubit, if our unknown is a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and our other bit to be copied to is $|0\rangle$ then our total input state is $\alpha|00\rangle + \beta|10\rangle$. If this is then our input state into a CNOT (Figure 2.3), we receive $\alpha|00\rangle + \beta|11\rangle$. This is not a copy of our original state, if we did receive a copy of our original state that was reversible we would find:

$$|\psi\rangle|\psi\rangle = \alpha^2|00\rangle + \alpha\beta|01\rangle + \alpha\beta|10\rangle + \beta^2|11\rangle \tag{2.31}$$

but we can see that terms of $\alpha\beta$ are missing. This is known as the *no-cloning theorem* and was first proven by Wootters and Zurek in 1982. [45].

Despite not being able to clone a qubit, our output from this operation is still useful. You may notice that it looks similar to out 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 2.7.

2.6. Quantum Algorithms and Parallelism.

2.6.1. *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, [46, 47] the Deutsch-Jozsa algorithm is one of the earliest examples of a quantum algorithm that outperforms its classical counterpart and while it doesnt have much practical use, it clearly shows the parallelism inherent in quantum

Input xy	Output z
$\|00\rangle$	$\|\Phi^+\rangle$
$\|01\rangle$	$\|\Phi^-\rangle$
$\|10\rangle$	$\|\Psi^+\rangle$
$\|11\rangle$	$\|\Psi^-\rangle$

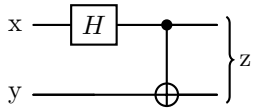

FIGURE 2.7. Circuit for creating a Bell state

computing.

The problem that the algorithm aims to solve is as follows: We have a quantum computer (U_f) of which we have no knowledge of its internal workings. It implements some function $f : \{0,1\}^n \to \{0,1\}$ which takes n-bit binarary 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 2.8.

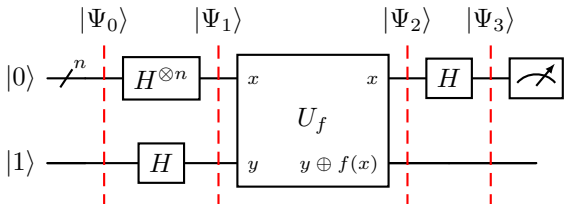

FIGURE 2.8. Circuit for the Deutsch-Jozsa algorithm

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 $|0\rangle^{\otimes n} |1\rangle$.
(2) We then apply the Hadamard gate to all qubits. This gives us the state:

$$|\Psi_1\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle \left(|0\rangle - |1\rangle\right)$$

Since x runs from 0 to $2^n - 1$, this is a superposition of all possible inputs.

(3) We then apply the function U_f to the state $|\Psi_1\rangle$, mapping input state $|x\rangle|y\rangle$ to $|x\rangle|y \oplus f(x)\rangle$. This gives us the state:

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle \left(|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle\right)$$

(4) Since we know that the output of U_f is either 0 or 1 for any input, so we can rewrite $|\Psi_2\rangle$ as

$$|\Psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle \left(|0\rangle - |1\rangle\right)$$

And we aren't interested in the last qubit so we can ignore it and rewrite the state as $|k\rangle$:

$$|k\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle$$

(5) We then apply the Hadamard gate to the first n qubits. This gives us the state:

$$H^{\otimes n}|k\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot y} |y\rangle$$

(6) Finally, we measure the n qubits:

$$\langle k|k\rangle = \left| \frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{f(x)} (-1)^{x \cdot k} \right|^2$$

The probability of measuring $|k\rangle = |0\rangle^{\otimes n}$ is 1 if $f(x)$ is constant and 0 if $f(x)$ is balanced. This is because if $f(x)$ is constant, then $(-1)^{f(x)}$ is the same for all x and so the sum is 1. If $f(x)$ is balanced, then $(-1)^{f(x)}$ is opposite for half the x and so the sum balances at 0.

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 the simplest example of a quantum algorithm that outperforms its classical counterpart.

2.6.2. *Bernstein-Vazirani Algorithm.*

2.6.3. *Simon's Algorithm.*

2.6.4. *Shor's Algorithm.*

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:

$$(m^e)^d \equiv m \mod n \tag{2.32}$$

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:

$$c \equiv m^e \mod n \tag{2.33}$$

$$m \equiv c^d \mod n \tag{2.34}$$

where c is the cipher text.

The RSA algorithm is said to be secure because breaking it requires recovering m such that Equation (2.33) 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.

2.6.5. *Grover's Algorithm.*

2.7. Quantum Error Correction.

2.8. Experimental Quantum Computing.

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* A400 (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. ; Mooij, 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. ; Lauigan, 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] Aaronson, Scott: *Quantum Computing since Democritus*. Cambridge University Press, 2013. – 132 S. https://doi.org/10.1017/CBO9780511976667

[32] 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

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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

[39] 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

[40] Copi, C. J. ; Cohen, M. ; McMahon, S.: Introduction to Logic. (2011). http://dx.doi.org/10.4324/9781315510897. – DOI 10.4324/9781315510897. ISBN 9781315510897

[41] 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

[42] 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

[43] 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

[44] Hadamard, J.S.: Résolution d'une question relative aux déterminants. In: *Bulletin des Sciences Mathématiques* 17 (1893), S. 240–246

[45] 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

[46] 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

[47] 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