Superconducting Qubits: A bibliographic review

1 Qubits Gates and Cirquits

1.1 Superposition

A fundamental tenet of quantum physics called superposition holds that a quantum system can exist in several states at once. A quantum particle, such an electron or a qubit, can exist in numerous locations, spin states, or energy levels simultaneously as a result. This feature of quantum systems, which is a crucial element of quantum computing and quantum cryptography, allows them to carry out specific sorts of computations much quicker than classical systems.

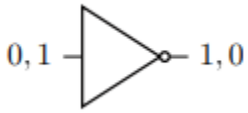

Figure 1: Classical NOT Circuit diagram

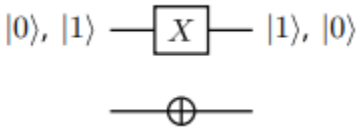

Figure 2: Quantum version of the NOT gate

1.2 No-cloning

A bit's state can be measured at any time and as many copies of the state can be made as desired in a classical logic circuit. If we know that a qubit is in one of the basis states, we can also perform this for it. The no Cloning theorem is a principle in quantum mechanics that states that it is impossible to make an exact copy of an arbitrary unknown quantum state. In other words, it is not possible to create a perfect replica of a quantum particle or a qubit and measure its properties without changing the original state. This property has important implications in quantum computing and quantum communication, as it ensures that quantum information cannot be duplicated or copied without leaving a trace, making it secure against certain types of hacking and eavesdropping. The No Cloning theorem is a fundamental aspect of quantum physics that sets it apart from classical physics and helps to explain some of the unique properties of quantum systems.

1.3 Reversibility

Reversibility is the capacity of a procedure or action to be reversed, allowing the system to revert to its initial condition. Reversibility in computing refers to the ability of the computation to be undone, turning the output back into the input. A key idea in computer science, reversible computing has significant applications in many disciplines, including quantum computing, cryptography, and error correction. In quantum computing, reversibility is important because it ensures that the quantum

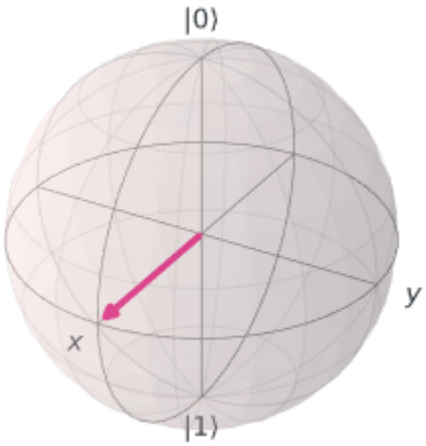

Figure 3: Representation of single qubit state

information stored in qubits is preserved and not lost during the computation process. This is achieved by using reversible logic gates, which are gates that can be undone and that preserve the quantum information stored in the qubits. Reversibility is also an important aspect of quantum algorithms, as it allows for the efficient manipulation of quantum information and the ability to solve complex problems in a more efficient manner than classical computing.

1.4 Entanglement

A quantum phenomenon known as entanglement occurs when two or more particles become so coupled that their states cannot be represented independently of one another, even though they are separated by a great distance. The particles are defined by a common wave function as a result of a phenomenon known as quantum entanglement, which creates this correlation. If two particles are entangled, the measurement of one particle will instantly affect the state of the other particle, regardless of the distance between them. This phenomenon is widely used in quantum computing and quantum communication, where entangled particles are used to create secure communication channels and perform computations that are faster and more efficient than classical computers. In other words, regardless of their distance from one another, if two particles are entangled, the measurement of one will immediately alter the state of the other particle.

1.5 Single-Qubit States

A single qubit can exist in a superposition of two states, represented by complex numbers, and is described by a vector in a two-dimensional Hilbert space. The two basis states of a qubit are often referred to as ”0” and ”1”. However, a qubit can be in any arbitrary state, represented by a linear combination of the basis states with complex coefficients, called amplitudes. The probabilities of measuring the qubit in the ”0” and ”1” states are given by the squares of the magnitudes of the amplitudes.

1.6 Born Rule and Measurement

Measurement in quantum mechanics is a process that collapses the state of a quantum system into one of its eigenstates, corresponding to the measurement outcome. The Born rule, is a fundamental tenet of quantum mechanics that expresses the likelihood that a measurement of a quantum system would produce a particular outcome.. The Born rule allows us to make predictions about the probabilities of the outcomes of quantum measurements and is a fundamental principle of quantum mechanics. After measurement, the state of the system becomes the eigenstate that was obtained, and any subsequent measurement of the same observable will always yield the same result.

1.7 Unitary Operations and Single-Qubit Gates and Two-Qubit Gates

A transition from one quantum state to another is known as a gate. Unitary operations that operate on a single qubit are known as single-qubit gates. They have the ability to modify a qubit's state and produce superpositions, entanglement, and other intricate quantum states. The Pauli X, Y, and Z gates, the Hadamard gate, the phase gate, and the rotation gates are typical single-qubit gates. These gates can be combined to create more intricate quantum circuits, which can then be utilized to execute quantum algorithms and calculations.On the other hand Two-qubit gates, also known as two-qubit operations or two-qubit unitaries, are unitary operations that act on two qubits simultaneously. They play an important role in quantum computing, as they allow for the creation of entanglement between two qubits and the implementation of quantum algorithms. The CNOT (controlled NOT) gate, the SWAP gate, the controlled-Z (CZ) gate, and the controlled-phase (CPHASE) gate are some popular two-qubit gates. The CNOT gate is particularly important because it allows the creation of entanglement between two qubits. One of the input qubits serves as the control and the other as the target for this gate. The target qubit is not changed if the control qubit is zero; nevertheless, the target qubit is flipped if the control qubit is one. .Moreover the SWAP gate is used in order to exchange the state between two qubits.

1.8 Bell states

The Bell states are a set of four maximally entangled quantum states that are widely used in quantum information science.A way in which bell states can be created is by using a combination of single-qubit operations and two-qubit gates, such as the CNOT gate. The creation of Bell states typically involves initializing both qubits in the —0⟩ state and then applying a series of gates to create the desired maximally entangled state.

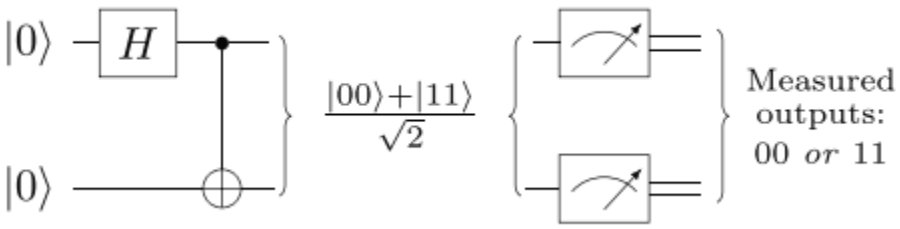

Figure 4: Cirquit For creating the bell state

2 Physics of Qubit Gates

2.1 Single Qubit Gates

Single qubit gates are a fundamental concept in quantum computing and quantum information science. These gates represent unitary transformations that are performed on a single quantum bit (qubit) to manipulate its quantum state. The state of a qubit is described by a two-dimensional complex vector and can be visualized as a point on the Bloch sphere. Single qubit gates rotate this point on the sphere to produce a desired state. The most commonly used single qubit gates include the Pauli-X, Pauli-Y, Pauli-Z, and Hadamard gates, each of which can be described by a unitary matrix. The combination of single qubit gates and multi-qubit gates is essential for performing quantum algorithms and simulations, and the study of these gates is a crucial area of research in the field of quantum computing and information science. rotation

2.2 Two Qubit Gates

Two qubit gates are operations performed on two qubits in quantum computing. These gates are essential for implementing multi-qubit quantum algorithms and simulations, and are a crucial component of

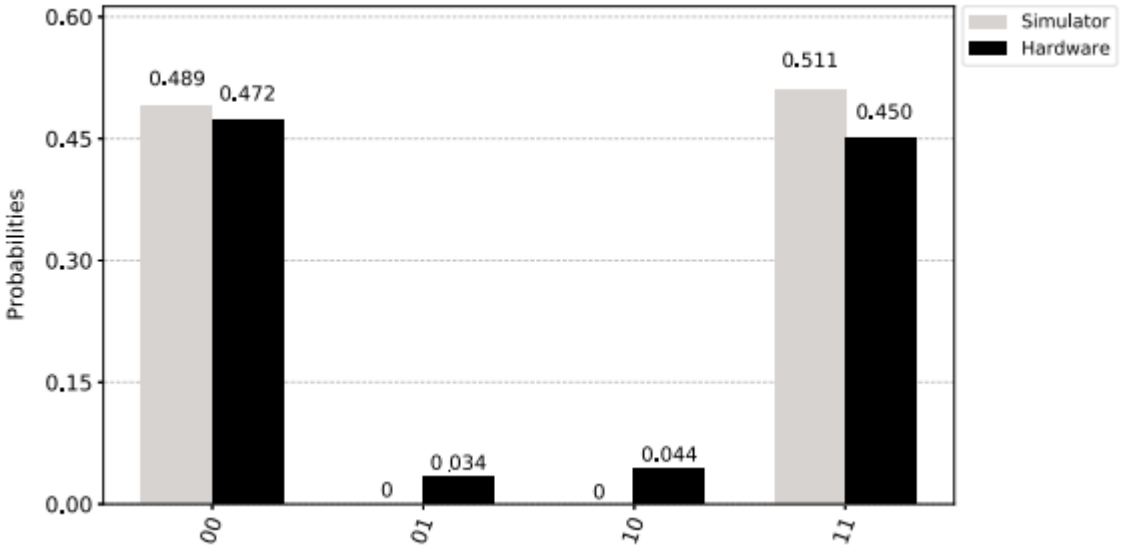

Figure 5: Result of executing the circuit 1024 times on a quantum simulator, compared with executing the circuit 1024 times on a real IBM quantum computer.

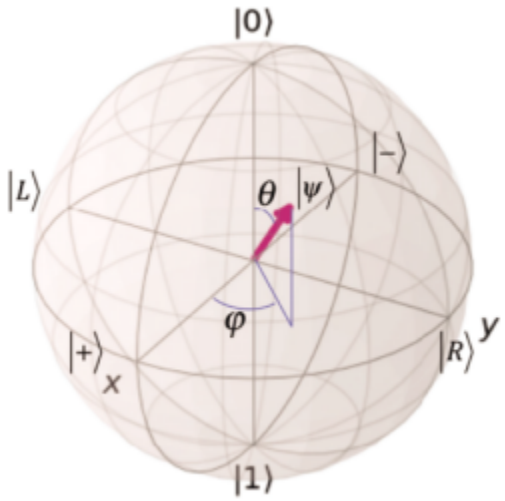

Figure 6: Representation of a single qubit state on the Bloch Sphere

quantum information processing. On this type of gates the operations are performed on the combined state of two qubits as opposed to single qubit gates, which change the state of a single qubit. This enables the development of entangled states and the implementation of quantum algorithms that are inefficiently solved by traditional computing techniques. The Controlled-NOT (CNOT), Controlled-Z (CZ), and SWAP gates are a few examples of typical two qubit gates. Unitary matrices can be used to describe the behavior of these gates, and rotations in a high-dimensional state space can be used to illustrate it.

2.2.1 Coupled Tunable Qubits

Coupled tunable qubits are a type of qubits in quantum computing that are designed to be easily coupled and controlled. In quantum computing, qubits are the basic building blocks for storing and processing quantum information, and the ability to couple qubits is crucial for implementing multi-qubit quantum algorithms and simulations. Coupled tunable qubits are designed such that the coupling

between the qubits can be controlled and adjusted, allowing for the manipulation of the joint state of the qubits. They can be implemented using various technologies, such as superconducting circuits, trapped ions, and nitrogen-vacancy centers in diamonds.

3 Quantum Error Correction

3.1 Definition of logic qubits

Qubit encoding is a need for quantum computing. Multiple quantum algorithms created in recent years have operated under the assumption that these qubits are flawless, able to be manufactured in any state, and controlled with absolute precision. Logic qubits are often referred to as qubits that adhere to these presumptions.

3.2 Definition of physical qubits

Although in the past years, the creation of qubits that bear more similarities to logic qubits is constantly evolving, the flaws will never vanish completely. Those kinds of qubits are called physical qubits.

3.3 Types of quantum errors

Several errors can occur in quantum computing systems, the most significant being bit-flip errors, phase-flip, and errors either from inaccurate handling of gate operations or state preparation, or because of contact with other qubits or the environment. The former error is called a classical bit flip, where a |0> becomes a |1> and vice versa. If we detect it, then we can easily correct it with an X gate. Phase-flip errors change the relative phase in the quantum state. This inaccuracy is not obvious while evaluating the computational base, yet phase differences can be crucial for the correct operation of many quantum algorithms. When found, this issue is similarly quickly fixed by using a Z gate. The latter type of error could appear hard to rectify, because we would need to precisely figure the rotation on each basis. The error measurement, however, will transform the current state into a different one that shows a bit flip or phase flip error, which later may be fixed.

3.4 Why quantum errors are more complicated?

In classical computing, errors are not missing as well, but quantum ones are far more complicated. One reason is that the quantum state seems to be much more complicated than a binary 1 or 0. Finally, the no-cloning theorem precludes us from making a copy of the state in order to measure it. Measuring the qubit to discover the mistake is damaging, as well. To solve this problem, we have created unique algorithms which are called quantum error correction codes.

3.5 Concept of quantum error correction codes

The basic idea is to encode the quantum information into a redundant form, such that we can recover the original information even in the presence of errors. In a QECC, we split the quantum information into multiple parts and encode them repetitively. Hence a change in one of the parts can be corrected by the others. QECCs also employ techniques such as syndrome measurement, which allows the detection of errors without destroying the quantum information. By using these techniques, it is possible to maintain the accuracy of the quantum information, even in the presence of errors.

3.6 Introduction to the repetition code

A repetition code typically consists of the data to be sent or stored. Then, the message is repeated to protect the data from noise. The data is finally decoded, which reduces the impacts of noise by putting your trust in the majority of transmitted signals.

3.7 Bit-flip

We will now discuss the bit flip error, in which we need to create a QEC code for a single logical qubit that will detect and correct it. We use |000>= |0> to represent a logical |0> and |111>= |1> to represent a logical |1>. An entangled superposition of our logical qubits:

$$|\psi> = \alpha|000> + \beta|111> = \alpha|0> + \beta|1> \tag{1}$$

is created from the initial state:

$$|\psi> = \alpha|0> + \beta|1> \tag{2}$$

as shown on the left of Figure 1. In mathematical terms, with a 2^3-dimensional space made using three physical qubits, we are embedding the two-dimensional Hilbert space that represents the single qubit state into a bigger Hilbert space. Only one physical qubit's bit flip can be our error operator E, meaning:

$$E\epsilon\{I, X_0, X_1, X_2\} \tag{3}$$

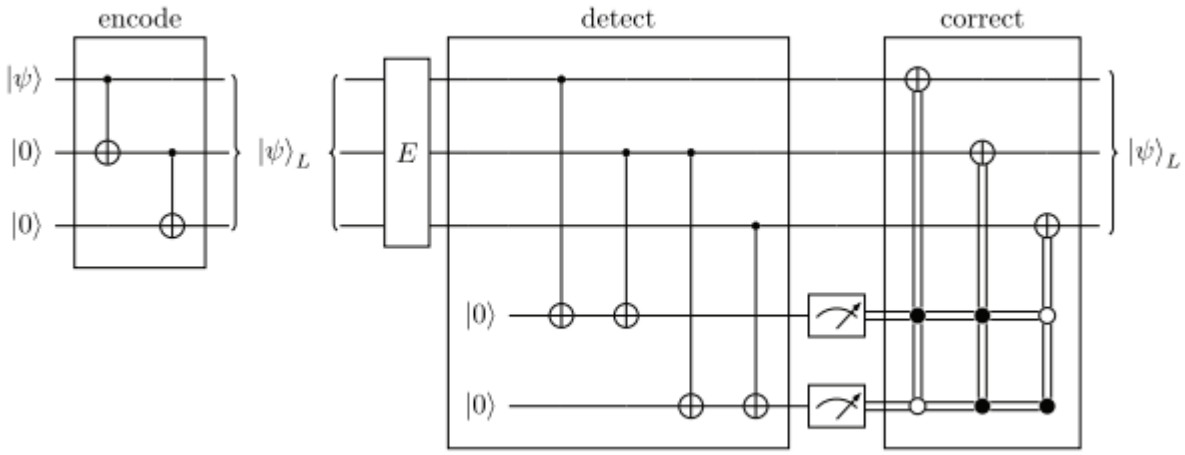

Figure 7: The circuit for three qubit bit flip error correction. On the left, the entangling of the qubits is shown. The detection circuit is shown on the middle. The double lines represent the classical bits after measurement. Closed circles are controls and the open circles are the targets of the CNOT gate.

Now that the data is encoded into a larger system, we must identify the bit flip. To do this, we look to see whether any bits change from the others. To repair the problem by using the X operator solely on the incorrect bit, which particular bit is altered must be established. We employ ancilla qubits to prevent direct measurement of data qubits, as seen in Figure 1.

If

$$|\psi>_0 and |\psi>_1 \tag{4}$$

are different, the first ancilla qubit q_0 will change to |1>. If not, it will always be |0>. In the same way, q1. We can establish which data qubit, whether any, has been flipped by evaluating these ancilla qubits, which yield two classical bits. The correction is displayed on the figure's right side, using the measured syndrome bits as controls. One of the data qubits, at most, is subjected to an X gate, which returns it to its initial logical state.

What happens in the case we have two-bit flips? The detection circuit will incorrectly categorize the fault, just like in the traditional scenario. As an illustration, the effect of applying an error of X1X0 to the input state of |000> is |011>. The corrected logical state of this error will be |111>, and it will be perceived as an error in qubit 2. To put it in another way, a logical bit-flip mistake will happen if there are two physical bit-flip faults.

This code may detect bit-flip errors but is inadequate for phase-flip errors. Consider the example with the initial state

$$|\psi> = \alpha|0> + \beta|1> \tag{5}$$

Any single bit can be phase-flipped to result in

$$\alpha|000> - \beta|111> \tag{6}$$

The phase error will not be detected since the detection circuit will output syndrome bits 00.

3.8 Phase-flip

To deal with phase-flip errors, we require a different strategy. We achieve that by adding a Hadamard gate after the encoding circuit, as seen on the left side of Figure 2. As a result, the encoding is |0>= |+++> and |1>= |—>. Our error operator can only observe single-qubit phase flips, in other words:

$$E\epsilon\{I, Z_0, Z_1, Z_2\} \tag{7}$$

A single-phase flip converts one of the |+> states into |-> or vice versa.

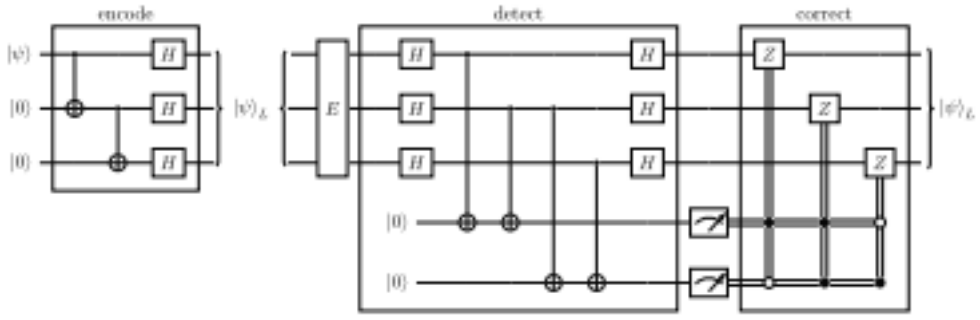

Figure 8: 2 Three-qubit error correction code for phase flips. On the left side, we see the encoding of the initial state and the application of Hadamard gates in each qubit.

Ancilla qubits are utilized by the error detection circuit to check if the relative phases of two qubits are equivalent. Hadamard gates are added in the circuit seen in Figure 2 to convert phase-encoded data qubits back into the computational basis. After creating syndrome bits with CNOT gates, the +/- basis is recovered with a new set of Hadamards. Subsequently, controlled Z gates are given to the flipped bit to apply a Z operator. Likewise, with the bit-flip circuit, the phase-flip cannot detect any other error except the phase-flip. Consider the case where we have the state

$$|\psi> = \alpha|0> + \beta|1> \tag{8}$$

Take a single physical qubit, like X0, that has a bit flip error. The resulting state is

$$\alpha|0> - \beta|1> \tag{9}$$

no correction is applied to the syndrome bits, which are computed as 00.

3.9 Shor's code

Now that we have seen bit-flip and phase-flip error correction codes, let's move to a QEC code which handles both. This is Shor's code which encodes a single qubit of information in a block of nine qubits, using the redundancy technique mentioned above. This allows the system to detect and correct errors that may occur during computation, even if they affect multiple qubits.

While the three groups of qubits (1,2,3), (4,5,6), and (7,8,9) are intended for the bit flip code, the first, fourth, and seventh qubits are used for the sign flip code. In the event of a bit flip error, the syndrome analysis is run on each block of qubits to find and fix no more than one-bit flip error per

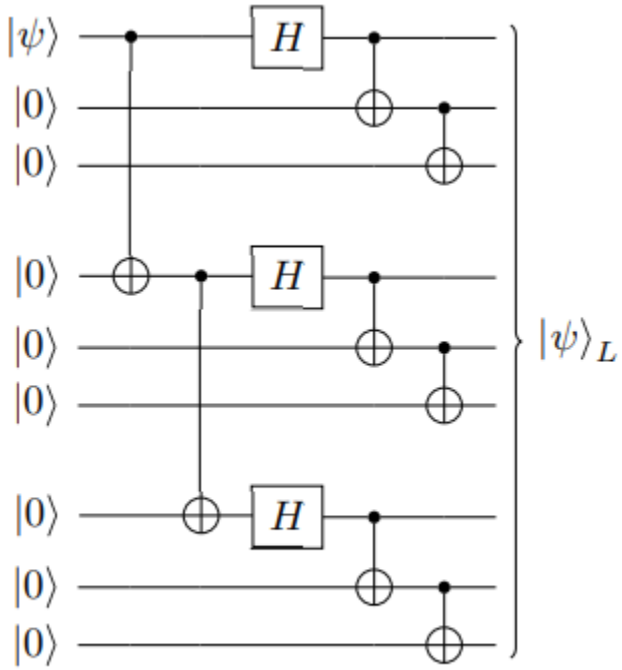

Figure 9: The circuit for Shor's code

block. The Shor code circuit can be simplified to a sign flip code, which means that it can also correct a sign flip error for a single qubit if the three-bit flip groups are thought of as three inputs.

Therefore, the Shor code can correct for any arbitrary errors (both bit flip and sign flip) to a single qubit. This makes it a powerful tool for building a reliable and scalable quantum computer, as it allows the system to detect and correct errors even if they affect multiple qubits.

3.10 Applications of QEC

Quantum error correction is required in many different disciplines. The creation of robust, large-scale quantum computers specifically depends on QEC. To safeguard quantum data in quantum memory and quantum processors, QEC is applied. Quantum entanglement communication (QEC) can also be utilized to increase the security and dependability of quantum communication systems like quantum key distribution. Last but not least, the stability and lifespan of quantum memory systems, such as trapped ion memories and superconducting quantum memories, are improved with the use of QEC in quantum storage.

In quantum computing, errors can occur as often as in classical computing. The nature of quantum computers requires innovative approaches to detect and correct these errors. Quantum error correction codes are designed to ensure that the accuracy of quantum information is maintained, even in the presence of errors. QECCs employ techniques such as syndrome measurement, which allows the detection of errors without destroying the quantum information.

4 Classical computation on Quantum Circuits

4.1 Reversibility

Classical computations can be implemented with quantum gates. In order for this to happen every operation that will be implemented in qubits would have to be reversible,meaning that when having the output it is possible to reproduce the input. Therefore, the gates used should be reversible. For a logic circuit that is combinational the 3 factors related to reversibility are:

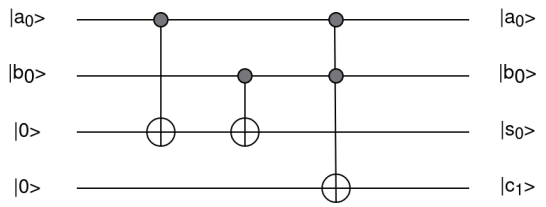

Figure 10: Quantum Half Adder Circuit

- input can be computed from the output and vice versa, meaning that the number of input and output signals is the same
- lack of any loops
- no explicit fan-out

An important issue is that in classical circuits fan-out can be simulated by CNOT gates in a way that is not permitted in quantum circuits. In order to make the circuit reversible we use ancilla bits. They can be initialized to zero or one and carry intermediate results or the output. When they are not needed they must be restored reversibly and not just reset. This process could allow ancilla bits to be reused in large circuits decreasing the need for more bits and complexity.

4.2 Quantum circuits considerations

After examining reversibility, basic logic and arithmetic computations can be performed with caution on special issues. Entanglement has no analog in classical circuits. Quantum inputs can be superpositions, therefore after the execution qubits may be entangled with each other. This could be a good and useful incedent depending on the state produced. During the execution, however, when entanglement happens in temporary values(ancilla bits) it can be a lot more complicated, risking affecting the state of a primary output qubit. This is where uncomputing discussed above is even more important.
Aditionally, today's quantum computers have some limitations. The implementations are usually simpler avoiding multi-qubit gates. This is because uncomputing ancilla bits in complex circuits leads to even more gates being used.
Qubit topology also introduces extra complexity to the design of a quantum system, when qubits need to be physically-connected, since this connection is a challenge by itself. Fortunately, in programming environments, operations like "SWAP" can be helpful in assuming a fully connected topology and consider connections part of the compilers responsibilities.

4.3 The Adder example

Starting of, the simplest form of an adder would be an one-bit adder with no carry input, known as a half adder. It is assumed that this arithmetic process refers to unsigned binary integers, for simplicity. The quantum implementation of such a circuit can be seen in picture 1. It requires the use of CNOT gates.

As an explanation, the possible cases are:

- both |a>, |b> are |0>, therefore all CNOT results in |0> for both outputs
- both |a>, |b> are |1>, meaning that |s0> will first flip to |1> and then again to |0> while |c1> will be flipped to |1>, as carry-out
- |a>, |b> are different and therefore |s0> will flip to |1> and |c1> will be flipped to |0>, as carry-out

Full-adder implementation is obviously more complex for the computation of higher bits, using ancilla bits and more CNOT gates. The carry out result will be used as input for a next step, using an extra CNOT. This is obvious looking at Figure 2 and comparing it to previous Figure 1. This

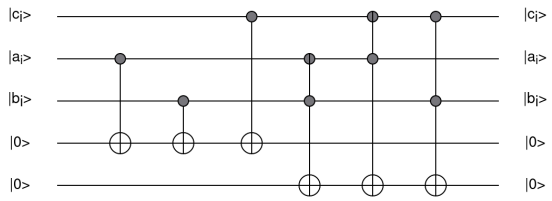

Figure 11: Quantum Full Adder Circuit

full-adder design can be replicated in order to create a n-bit adder circuit, each time treating the carry-out bit as a next input. These carry-out bits that keep intermediate results and pass them on are a great example of ancilla bits in use. As explained before they need to be used again or reclaimed they would have to be uncomputed in opposite order in the circuit.

As in classic circuits the need to compare different implementation and achieve better results leads to a group of factors being characterized as essential in efficiency evaluation. To evaluate a quantum design one would have to measure the number of qubits used for the circuit as long as the number of gates and the final depth and size of the circuit. The final total number of gates expresses the size and the longest sequence gives the depth of a circuit. Input and output qubits are of course essential so it all comes down to the scarcity in which ancilla bits are employed for the computations. It is also worth mentioning, that when it comes to size and depth even two-qubit, and Toffoli gates count as a single qubit gate.

The circuits computing addition can be implemented in different ways for better efficiency. An improvement for instance could be the parallel computation of non-carry part of the sum. This way the depth of the circuit would decrease and parallelism in general can help in cases where the dependency on carry bits allows it. In other cases reordering is proven helpful to decrease this dependency and provide more flexibility without affecting the computations. Some classical optimization techniques include:

- constant propagation, by removing unnecessary constants with known values
- dead expression elimination, by avoiding expressions that are not needed for computations or incorporating them with useful ones
- dependency analysis, by experimenting with gates and timing in the arrangement of gates in a quantum circuit

Such standarnd rearrangements are often done by the compiler that tries to arrange gates in the best possible manner. This of course does not include algorithmic changes that are less likely and therefore fall under the programmers responsibilities.

4.4 Phase logic

Although logic in quantum circuits can operate in computational basis, this is not always the case. Phase logic allows qubits to interfere in different ways and has proven valuable in the implementation of many quantum algorithms. Special notation exists to help visualize the phase and magnitude of components in multi-qubit states and logic operations could help avoid ancilla bits by using the phase of the output to encode the result. This phase logic is implemented as the last step of circuits working in standard logic and it is followed by uncomputing.

5 Special issues in quantum circuits

5.1 Parallelism in quantum circuits

Existing parallelism techniques that have proven effective in classical architecture cannot be easily applied to quantum michroarchitecture. Different solutions have been proposed in order to achieve parallelism in quantum computing systems. This kind of parallelism can be divided in two main categories:

- Circuit Level Parallelism (CLP): Sub-circuits in bigger architectures should ideally be able to run in parallel to an extent.
- Quantum Operation Level Parallelism: Full operations could also be executed in parallel using different qubits.

It can be difficult for the above to be combined. For example, quantum feedback control in form of branching or extra transitional measurements can help operations but increases complexity and obstructs circuit level parallelism

5.2 Security issues in quantum circuits

While big companies are offering cloud-based access to physical quantum computer and physical hardware is constantly developed the issues concerning privacy and security for quantum circuits should be highlighted. Some of the main attacks are presented below as a summary of superconducting quantum circuits' vulnerabilities.

Cross talk fault injection In an environment where more than one quantum programs run simultaneously on different qubits, an attacker could potentially exploit cross talk error and launch fault-injection attacks. The significance of such attack can differ depending on the computational process type and importance. Buffer qubits between the programs have been proposed as a countermeasure to this attack.

Scheduler Attack Usually a provider allocates the hardware for the implementation of a quantum circuit. A new type of attack could exploit the fact that the user does not know this hardware information and direct the operations to inferior hardware. The attacker could also mess with the policy on the queue for quantum circuits.

Readout error exploitation Readout or measurement error as a phenomenon depends on the state of a qubit and when read qubits' states show asymmetric bit-flip probabilities. Imaginging a situation where qubits belong to two programs, an attacker could sense the state of a victim's qubit by:

- collecting information reading both qubits
- reading his own qubit and using the data collected making an accurate guess on the victim's qubit

Ancillary bits and garbage lines exploitation The reversibility explained above is consider an important factor for quantum circuits. This kind of circuits are considered to have better privacy protection but the synhtesis adds ancillary and garbage lines. These can be used as signs to understand the functionality of a circuit, giving an advantage to the attacker.

5.3 Towards safer quantum circuits

Concerning the above attacks many techniques have been utilized as countermeasures. For instance, the idea of buffer qubits between the programs has been proposed and tested as a countermeasure to cross talk fault injection. It is also advised that error measurements are tracked since targeted monitoring with known output could detect suspicious anomalies in error rate. When it comes to the quantum architecture dummy gates can aid in hiding the true functionality of the original circuit offering some privacy against an untrusted compiler.
At the same time, new theoretical protocols are focused in "blind quantum computation". This kind of process gives the ability to a client to communicate with a server and perform computations in a way that input, output and computations are kept completely private. Combining homomorphic encryption with quantum circuits provides such a scheme for processing hidden data.

References

Quantum gates and circuits
Elementary gates for quantum computation

Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies

Geometry of entangled states
Geometry of entangled states, Bloch spheres and Hopf fibrations
Principles of Superconducting Quantum Computers

Quantum Error Correction

Wikipedia (Born Rule)

Introduction to Quantum Error Correction using Repetition Codes

Qiskit: An Open-source Framework for Quantum Computing

Quantum computation and quantum information

An Introduction to Quantum Error Correction and FaultTolerant Quantum Computation. Apr

Multiple-Particle Interference and Quantum Error Correction
Efficient Computations of Encodings for Quantum Error Correction//
Quantum Error Correction for Beginners

Correcting quantum errors with entanglement

Realization of quantum error correction

Theory of quantum error-correcting codes

A Survey and Tutorial on Security and Resilience of Quantum Computing

Exploiting Different Levels of Parallelism in the Quantum Control Microarchitecture for Superconducting Qubits

Quantum Addition Circuits and Unbounded Fan-Out

Superconducting qubits: Current state of play

Digital quantum computation with superconducting qubits

Efficient Quantum Circuit Decompositions via Intermediate Qudits

Architectural implications of quantum computing technologies

Quantum Computing based Implementation of Full Adder

Reversible Adder Circuits (RAC) realization using Quantum Technologies

Circuit compilation methodologies for quantum approximate optimization algorithm