Quantum Computing: Designing and Simulating Quantum Circuits

Afonso Alemão
MEEC
Instituto Superior Técnico
Student Number: 96135
Lisbon, Portugal
afonso.alemao@tecnico.ulisboa.pt

Rui Daniel
MEEC
Instituto Superior Técnico
Student Number: 96317
Lisbon, Portugal
ruipcdaniel@tecnico.ulisboa.pt

Tomás Fonseca
MEEC
Instituto Superior Técnico
Student Number: 66325
Lisbon, Portugal
tomasfonseca@tecnico.ulisboa.pt

***Abstract*—In this work, we simulate simple quantum circuits, identifying the functions they compute, taking advantage of superposition and solving dilemmas. For that, we use *Quirk: Quantum Circuit Simulator*. We analyse, discuss, describe the attained results and draw the main conclusions for each of the simulated circuits: Pauli Gates, T Gate, Not Gate, CNOT Gate, Hadamard Gate, Toffoli Gate and Quantum Full Adder. For the Quantum Full Adder we also modify the circuit for taking advantage of the superposition, allowing the physical system to be in one of different configurations, corresponding to more than one state. For all circuits the results obtained in simulation are equal to the theoretical values.**

I. Introduction

All the referenced Figures and Tables are in the Attachments.

Quantum computation is a new type of reversible and invertible computation [1], [2], [3] [4]. The most basic piece of quatum data bit is a qubit. A qubit is a vector of two elements, and can represent the two states of a classical bit (values 0 and 1). These states can be represented as $|0\rangle$ and $|1\rangle$ in Dirac vector notation as presented in (1). Moreover, a quantum state may then be found in any quantum superposition $|\Psi\rangle$ of the two classical states $|0\rangle$ and $|1\rangle$. This is shown in (2).

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1}$$

$$|\Psi\rangle = \rho\,|0\rangle + \sigma\,|1\rangle = \begin{pmatrix} \rho \\ \sigma \end{pmatrix}, \quad \|\rho\|^2 + \|\sigma\|^2 = 1 \tag{2}$$

A qubit represents superposition since it is both 0 and 1 at the same time. Only when a measurement is performed in a quantum system, the qubit collapses to an actual value 0 or 1. The state $|\Psi\rangle$ can be observed as a probability vector. It collapses to zero state with probability equal to $\|\rho\|^2$ and it collapses to one state with probability equal to $\|\sigma\|^2$. $\|\rho\|^2$ and $\|\sigma\|^2$ are designated the quantum amplitudes.

The state of $|\Psi\rangle$ changes with the application of quantum gates that corresponds to linear algebra matrix multiplication. Matrix operators model the effect of some device which manipulates qubit spin/polarization without measuring and collapsing it.

In the next section we discuss some fundamental gates such as Hadamard gate, Not gate, C-NOT gate and Toffoli gate.

In particular, the Hadamard gate shows that it is possible to transition out of superposition without measurement and that it is possible to structure quantum computation deterministically instead of probabilistically.

When operating on one classical bit (cbit) we can use simple gates such as Identity and NOT that are reversible and Constant-0 and Constant-1 that are not reversible. However, quantum computers only use reverse operations and all quantum operators are their own inverses. Identity and NOT are valid operations, but Constant-0 and Constant-1, in order to become valid, need to be wrote in a reversible way: add an additional output qubit to which the function action is applied.

A quantum state can be represented in a unitary sphere called the Bloch Sphere that is represented in Fig. 1. In this case, a qubit is given by (3), for $\alpha, \beta, \varphi, \theta \in \mathbb{R}$.

$$|\Psi\rangle = \alpha\,|0\rangle + e^{i\varphi}\beta\,|1\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\varphi}\sin\left(\frac{\theta}{2}\right)|1\rangle \tag{3}$$

In the *Quirk Simulator* we can insert as input qubit the conventional pole states. The ones defined in the y-axis of Fig. 1 are given by (4) and the ones defined in the x-axis are given by (5).

$$|i+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + i\,|1\rangle\right), \quad |i-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - i\,|1\rangle\right) \tag{4}$$

$$|+\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \quad |-\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right) \tag{5}$$

In addition, multi-qubits can be derived by applying tensor algebra to single qubits as in $|00\rangle = |0\rangle \otimes |0\rangle$.

In fact, multi-bit states are written as the tensor product of single-bit vectors, except when quantum entanglement is registered. Quantum entanglement is a physical phenomenon that occurs when each particle of a group of particles cannot be described independently of the state of the others. If the product state of two qubits cannot be factored they are said to be entangled, as in (6).

$$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} \tag{6}$$

In quantum computing no transmission medium is needed.

The Deutsch Oracle proposes to solve the following problem. There are 4 possibilities for a black box (BB) containing a function of one bit: Constant-0, Constant-1, Identity and Negation. In order to write nonreversible functions in a reversible way it is added an additional output qubit to which the function is applied. The BB leaves the input qubit unchanged, writing the function output to output qubit, as in Fig. 2.

It is possible to determine whether the function is constant or variable in a quantum computer with a single query using the circuit in Fig. 3.

If the BB function is constant, the system will be in state $|11\rangle$ after measurement. On the other hand, if the BB function is variable, the system will be in state $|01\rangle$ after measurement.

The main goal of this project is to simulate simple quantum circuits, identifying the functions they compute, taking advantage of superposition and solving dilemmas.

II. METHODOLOGY

In this section we analyse and discuss the simulated circuits.

A. Quirk Simulator

Quirk is a quantum circuit simulator for the purpose of learning quantum computing. Its code can be found on *GitHub* [5]. *Quirk* runs in a web browser, providing us with drag-and-drop circuit editing. In addition, it allows real-time interaction, simulation and animation, and supports markable and linkable circuits. The main limitation of the simulator is that the maximum number of qubits supported is 16 qubits.

To become familiar with the *Quirk* operation and usage, we design simple quantum circuits, using the studied gates.

B. Simple Quantum Gates

1) Pauli Gates, Phase Gates and T Gate: Pauli gates are a set of quantum logic gates named after the 3 axis of Bloch Sphere. Each gate applies a rotation of π around the axis that names it. They can be represented by the matrix operators in (7).

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}.\ Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{7}$$

By multiplying the Pauli matrices with a single-qubit input, the theoretical results, represented in Tables I, II and III are obtained.

We can also define a Phase gate: a quantum gate that applies a rotation around the z-axis, that is represented by the matrix operator $P(\varphi)$ in (8).

In addition to this, we can also define a T gate: a quantum gate that applies a fixed rotation around the z-axis, that is represented by the matrix operator T in (8) [6]. It is also possible to define the conjugate transpose of the T gate by $T^\dagger$ in (8) and its representation in *Quirk* is T^{-1}.

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix},\ T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{bmatrix},\ T^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\frac{\pi}{4}} \end{bmatrix} \tag{8}$$

By multiplying the T matrix with a single-qubit input, the theoretical results, represented in Table IV, are obtained.

2) Not Gate: In quantum computing, a Not gate (also called a Pauli-X gate) is a quantum logic gate that performs the logical negation operation. The Not gate can be represented by the matrix operator X in (7).

By multiplying the NOT matrix operator with a single-qubit input, the theoretical results, represented in Table V, are obtained.

3) Hadamard Gate: In quantum computing, the Hadamard gate is a single-qubit quantum gate that returns a quantum state that is equal to a phase change in the original state. It is defined by the matrix operator H in (9).

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{9}$$

The Hadamard gate acts on a single qubit and has the effect of a $\pi/2$ radians rotation around the y-axis, followed by a π radians rotation around the x-axis in the Bloch Sphere representation. So, it is possible to represent H as in (10) [7].

$$H = XY^{\frac{1}{2}} \tag{10}$$

By multiplying the H matrix operator with a single-qubit input, the theoretical results, represented in Table VI, are obtained.

The operation of the Hadamard gate can be visualized as follows: if the qubit is in the state $|0\rangle$, applying the Hadamard gate will rotate the state to the state $|+\rangle$, which is a superposition of $|0\rangle$ and $|1\rangle$. If the qubit is in the state $|1\rangle$, applying the Hadamard gate will rotate the state to the state $|-\rangle$, which is also a superposition of $|0\rangle$ and $|1\rangle$.

So, this gate takes a 0 or 1 bit and puts it into exactly equal superposition. It also takes a qubit in exactly equal superposition and transforms it into a 0 or 1 bit.

C. CNOT Gate

In quantum computing, the controlled NOT gate (CNOT gate) is a fundamental building block of reversible computing.

The CNOT gate is a two-qubit gate that performs a NOT operation on the target qubit if and only if the control qubit is in the one state. The control qubit is always unchanged. Formally, the action of a CNOT gate on a two-qubit state $|c, t\rangle$, where $|c\rangle$ is the control qubit and $|t\rangle$ is the target qubit can be written as in (11). The CNOT gate is represented by the matrix operator in (12).

$$\text{CNOT}\,|c, t\rangle = |c,\ c \text{ XOR } t\rangle \tag{11}$$

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{12}$$

By multiplying the CNOT matrix operator with a two-qubit input, the theoretical results are obtained. They are represented in Table VII where $|c\rangle$ and $|t\rangle$ are respectively the control and target qubits.

Taking advantage of the superposition, we can find the equivalent circuit for the CNOT gate, represented on Fig. 4, using only simple quantum gates [8].

D. Toffoli Gate

A Toffoli gate is a three-qubit quantum gate that performs a reversible operation on three qubits that takes as input. Two of them serve as control and the other qubit serves as the target. If the control qubits are both in the one state, the Toffoli gate flips the state of the target qubit. If at least one of the control qubits are in the zero state, the Toffoli gate leaves the target qubit unchanged. The Toffoli gate is represented by the matrix operator in (13).

$$\text{Toffoli} = \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \end{bmatrix} \tag{13}$$

By multiplying the Toffoli matrix operator by a three-qubit input, the theoretical results are obtained. They are represented in Table VIII where $|c_0\rangle$ and $|c_1\rangle$ are the control qubits and $|t\rangle$ is the target qubit.

Taking advantage of the superposition, we can find the equivalent circuit for the Toffoli gate, represented on Fig. 5, using only simple quantum gates [9].

Formally, the action of a Toffoli gate on a three-qubit state $|c_0, c_1, t\rangle$, where $|c_0\rangle$ and $|c_1\rangle$ are the control qubits and $|t\rangle$ is the target qubit can be written as in (14).

$$\text{Toffoli}\,|c_1, c_0, t\rangle = |c_1, c_0,\ (c_0 \text{ AND } c_1) \text{ XOR } t\rangle \tag{14}$$

E. Quantum Full Adder

We analyse the quantum circuit in Fig. 6 [10].

First, we get this logical function expressions for the outputs. The symbol $\oplus$ means XOR.

In fact, for the circuit presented in Fig. 6, we make use of the CNOT and Toffoli gates. As it was discussed before, applying a CNOT gate to 2 qubits, $|a_1\rangle$, $|a_2\rangle$, can be represented by the operation (11) represented in Fig. 7 where $|a_1\rangle$ is the control qubit and $|a_2\rangle$ is the target qubit.

As for the Tofolli gate, we verify in (14) that we can represent its operation by Fig. 8, where $|a_1\rangle$ and $|a_2\rangle$ are the control qubits and $|a_3\rangle$ is the target qubit.

Having this into account, the output logical functions can be given by (15), (16), (17) and (18).

$$y_0 = x_0 \tag{15}$$

$$y_1 = (x_1 \oplus x_0) \oplus x_0 = x_1 \tag{16}$$

$$y_2 = x_2 \oplus (x_0 \oplus x_1) \tag{17}$$

$$y_3 = (x_3 \oplus (x_0 \wedge x_1)) \oplus (x_2 \wedge (x_1 \oplus x_0)) \tag{18}$$

In order to write the circuit function in a reversible way, we use $|x_3\rangle$ as an additional output qubit to which the function action is applied. Having this said, $|x_3\rangle$ is set to $|0\rangle$, meaning x_3 set to 0, and the output y_3 can now be expressed as in (19).

$$y_3 = (x_0 \wedge x_1) \oplus (x_2 \wedge (x_1 \oplus x_0)) = (x_0 \wedge x_1) \vee (x_2 \wedge (x_1 \oplus x_0)) \tag{19}$$

Therefore, the logic function that the circuit performs can be represented in the diagram in Fig. 9 based on (15), (16), (17) and (19).

By that diagram, it is possible to conclude that this quantum circuit is a Quantum Full Adder, where $|x_0\rangle$, $|x_1\rangle$, $|x_2\rangle$ and $|x_3\rangle$ are given by (20) and $|y_0\rangle$, $|y_1\rangle$, $|y_2\rangle$ and $|y_3\rangle$ are given by (21).

$$|x_0\rangle = |A\rangle\,,\ |x_1\rangle = |B\rangle\,,\ |x_2\rangle = |C_{in}\rangle\,,\ |x_3\rangle = |0\rangle \tag{20}$$

$$|y_0\rangle = |A\rangle\,,\ |y_1\rangle = |B\rangle\,,\ |y_2\rangle = |S\rangle\,,\ |y_3\rangle = |C_{out}\rangle \tag{21}$$

C_{in} is a short for Carry in, as it carries in the qubit from the previous Full Adder since these can be stringed together. S is the sum of inputs A and B. C_{out} is a short for Carry out as it carries out a qubit to the C_{in} of the next Full Adder.

Then, using (15), (16), (17) and (19) we get the theoretical results for the circuit operation, by changing the state of the input quantum bits and registering the outputs. The results are reported in Table IX.

It is possible to write the input and output multi-qubits as $|x\rangle \in \mathbb{R}^{16}$ and $|y\rangle \in \mathbb{R}^{16}$ by (22) and (23). From there, we achieve the matrix M in (24) that represents the circuit.

$$|x\rangle = |x_3 x_2 x_1 x_0\rangle = |x_3\rangle \otimes |x_2\rangle \otimes |x_1\rangle \otimes |x_0\rangle \tag{22}$$

$$|y\rangle = |y_3 y_2 y_1 y_0\rangle = |y_3\rangle \otimes |y_2\rangle \otimes |y_1\rangle \otimes |y_0\rangle \tag{23}$$

$$M = \begin{bmatrix}
\mathbf{1}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&\mathbf{1}&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&\mathbf{1}&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&\mathbf{1}&0&0&0&0\\
0&0&0&0&\mathbf{1}&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&\mathbf{1}&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&\mathbf{1}&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\mathbf{1}\\
0&0&0&0&0&0&0&0&\mathbf{1}&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&\mathbf{1}&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&\mathbf{1}&0\\
0&0&0&\mathbf{1}&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&\mathbf{1}&0&0&0\\
0&\mathbf{1}&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&\mathbf{1}&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&\mathbf{1}&0&0&0&0&0&0&0&0
\end{bmatrix}, \quad |y\rangle = M\,|x\rangle \tag{24}$$

Another way to get the boolean expressions for these outputs is by using Karnaugh maps. Since the logical functions for y_0 and y_1 are trivial, given by (15) and (16), we proceed to build the Karnaugh maps for y_2 and y_3, represented in Fig. 10 and Fig. 11. Through these, we get the simplified boolean expressions for y_2 and y_3 in (25) and (26). In fact, (25) and (26) are equivalent to (17) and (19), respectively.

$$y_2 = \bar{x_2}\bar{x_1}x_0 + \bar{x_2}x_1\bar{x_0} + x_2\bar{x_1}\bar{x_0} + x_2 x_1 x_0 \tag{25}$$

$$y_3 = x_3\left(\bar{x_2}\bar{x_1} + \bar{x_2}\bar{x_0} + \bar{x_1}\bar{x_0}\right) + \bar{x_3}\left(x_1 x_0 + x_2 x_0 + x_2 x_1\right) \tag{26}$$

To sum up, since quantum circuits are reversible, they have an equal amount of input and output qubits. Therefore, we define a 4-qubit function represented in Fig. 12, where the input qubits are $|A\rangle$, $|B\rangle$, $|C_{in}\rangle$ and $|0\rangle$ and the output qubits are $|A\rangle$, $|B\rangle$, $|S\rangle$ and $|C_{out}\rangle$.

Now we describe the computation of $|S\rangle$. First, we apply a CNOT with control qubit $|A\rangle$ and target qubit $|B\rangle$ and then use this result as the control qubit of a new CNOT gate whose target qubit is $|C_{in}\rangle$.

This means that if the sum of A, B and C_{in} is equal to 0, then $S = 0$ and $C_{out} = 0$. If the sum of A, B and C_{in} is equal to 1, then $S = 1$ and $C_{out} = 0$. If the sum of A, B and C_{in} is equal to 2, then $S = 0$ and $C_{out} = 1$. Otherwise, if the sum of A, B and C_{in} is equal to 3, then $S = 1$ and $C_{out} = 1$.

Then, we describe how to compute $|C_{out}\rangle$. First, we apply a Toffoli gate with control qubits $|A\rangle$ and $|B\rangle$ and target $|0\rangle$. Afterwards, we apply another Toffoli gate with target qubit equal to the result from the previous Toffoli gate operation. The control qubits of this final Toffoli gate are given by $|C_{in}\rangle$ and by the result of a CNOT with control qubit $|A\rangle$ and target qubit $|B\rangle$.

Changing topic, we define superposition as qubit both 0 and 1 at the same time. When its measured it collapses to an actual value 0 or 1. The Hadamard gate allows the transition out of superposition without measurement.

We can modify the circuit for taking advantage of the superposition, allowing the physical system to be in one of different configurations, corresponding to more than one state. For that, we use the equivalent circuits for CNOT gate and for Toffoli gate that take advantage of the superposition, in Fig. 4 and Fig. 5, respectively.

By replacing the Toffoli gates and some of the CNOT gates by their equivalent circuits in Fig. 6 and also by simplifying the successive reversible gates (e.g., applying two sucessive H gates to a qubit is equivalent to applying an Identity gate, because all quantum operators are their own inverses), we obtain the circuit in Fig. 13. The CNOT gates were simplified in the case of leading to simplification of successive reversible gates.

The results are the same that for the original circuit, represented in Table IX.

III. Results

For each quantum circuit we describe the attained results obtained by *Quirk Simulator* and draw the main conclusions. We simulate each circuit by changing the state of the input quantum bits and registering the outputs.

A. Simple Quantum Gates

1) Pauli Gates: We simulate the Pauli gates operation in Fig. 14, Fig. 15 and Fig. 16. The simulations can be found at [11], [12] and [13]. The results obtained in simulation are reported in Tables X, XI and XII. They are equal to the theoretical values in Tables I, II and III.

2) T Gate: We simulate the T gate operation in Fig. 17. The simulation can be found at [14]. The results obtained in simulation are reported in Table XIII. They are equal to the theoretical values in Table IV.

3) Not Gate: We simulate the Not quantum gate operation in Fig. 14. The simulation can be found at [11]. The results obtained in simulation are equal reported in Table X. They are equal to the theoretical values in Table V.

4) Hadamard Gate: We simulate the Hadamard gate operation in Fig. 18. You can find the simulation in [15]. The results obtained in simulation are reported in Table XIV. They are equal to the theoretical values in Table VI.

B. CNOT Gate

We simulate the CNOT gate operation in Fig. 19. The simulation can be found at [16]. The results obtained in simulation are reported in Table XV where $|t\rangle$ and $|y_t\rangle$ are the qubits represented in the line further down in the *Quirk Simulator*. They are equal to the theoretical values in Table VII.

C. Toffoli Gate

We simulate the Toffoli gate operation in Fig. 20 and in Fig. 21. You can find the simulations in [17] and [18]. The results are reported in Table XVI where $|t\rangle$ and $|y_t\rangle$ are the qubits represented in the line further down in the *Quirk Simulator*. $|c_0\rangle$ and $|c_1\rangle$ are the control qubits and $|t\rangle$ is the target qubit. The results obtained in simulation are equal to the theoretical values in Table VIII.

D. Quantum Full Adder

Regarding the circuit represented in Fig. 6, we simulate its operation in Fig. 22. You can find the simulation in [19]. The results obtained are equal to the theoretical values in Table IX. They are reported in Table XVII where $|x_3\rangle$ and $|y_3\rangle$ are the qubits represented in the line further down in the *Quirk Simulator*.

We also use the matrix M to create a new gate in *Quirk* and we simulate the circuit operation in [20]. The results are equal to the ones reported in Table XVII that are equal to the theoretical values.

Then, we simulate the modified the circuit that takes advantage of the superposition, represented in Fig. 13. You can find the simulation in [21]. The results are equal to the ones reported in Table XVII that are equal to the theoretical values.

IV. Conclusions

In this project, we simulate simple quantum circuits using *Quirk: Quantum Circuit Simulator*, identifying the functions they compute, taking advantage of superposition and solving dilemmas.

We explain the basis of the quantum computing and the aim of the work. We analyse, discuss, describe the attained results and draw the main conclusions for each of the simulated circuits: Pauli Gates, T Gate, Not Gate, CNOT Gate, Hadamard Gate, Toffoli Gate and Quantum Full Adder.

For the Quantum Full Adder we also modify the circuit for taking advantage of the superposition, allowing the physical system to be in one of different configurations, corresponding to more than one state.

For all circuits the results obtained in simulation are equal to the theoretical values.

Acknowledgment

This work was done in the Technologies of Computing Systems course from Instituto Superior Técnico, Lisbon, teached by Professor Leonel Augusto Pires Seabra de Sousa and Professor Diogo Miguel Bárbara Coroas Prista Caetano.

References

[1] 1st Semester 2022/2023, Second Quarter 22/23, Technologies of Computing Systems, 2nd Project, Quantum Computing: Designing and Simulating Quantum Circuits, by Prof. Leonel Sousa, and Prof. Gonçalo Valentim.

[2] Technologies of Computing Systems Slides - Slides of Theoretical Classes 2022/2023, Quantum Computing, 1st Semester (MEEC), by Professor Leonel Sousa, with the collaboration of Professor Gonçalo Valentim.

[3] "Quantum Inspire: The basics of Quantum Computing". Available: https://www.quantum-inspire.com/kbase/introduction-to-quantum-computing/.

[4] Yanofsky, N., and Mannucci, M., "Quantum Computing for Computer Scientists", Cambridge University Press.

[5] *Quirk Simulator*. Available: https://github.com/Strilanc/Quirk.

[6] Quantum Inspire: T Gate. Available: https://www.quantum-inspire.com/kbase/t-gate/.

[7] Quantum Inspire: Hadamard Gate. Available: https://www.quantum-inspire.com/kbase/hadamard/.

[8] *Quantum Computing Stack Exchange*. Available: https://quantumcomputing.stackexchange.com/questions/13782/cnot-expressed-with-cz-and-h-gates-by-taking-into-account-hzh-x.

[9] Shende, V. V., and Markov, I. L., "On the CNOT-cost of TOFFOLI gates", arXiv preprint arXiv:0803.2316.

[10] Research Gate. Available: https://www.researchgate.net/figure/Circuit-model-of-a-quantum-full-adder_fig1_346508632.

[11] *Quirk - Quantum Circuit Simulator*: Quantum NOT Gate. Available: https://bit.ly/3Gt8KJo.

[12] *Quirk - Quantum Circuit Simulator*: Quantum Y Gate. Available: https://bit.ly/3Gtodcq.

[13] *Quirk - Quantum Circuit Simulator*: Quantum Z Gate. Available: https://bit.ly/3GSpV8E.

[14] *Quirk - Quantum Circuit Simulator*: Quantum T Gate. Available: https://bit.ly/3CAgQyC.

[15] *Quirk - Quantum Circuit Simulator*: Quantum Hadamard Gate. Available: https://bit.ly/3ioN3C6.

[16] *Quirk - Quantum Circuit Simulator*: Quantum CNOT Gate. Available: https://bit.ly/3QyTaAA.

[17] *Quirk - Quantum Circuit Simulator*: Quantum Toffoli Gate. Available: https://bit.ly/3QreK9W.

[18] *Quirk - Quantum Circuit Simulator*: Quantum Toffoli Gate taking advantage of superposition. Available: https://bit.ly/3XjiIE0.

[19] *Quirk - Quantum Circuit Simulator*: Quantum Full Adder. Available: https://bit.ly/3vVdmTq.

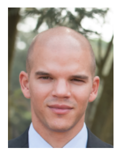
[20] *Quirk - Quantum Circuit Simulator*: Quantum gate using matrix M. Available: https://bit.ly/3GACHXI.

[21] *Quirk - Quantum Circuit Simulator*: Quantum Full Adder Taking Advantage of Superposition. Available: https://bit.ly/3CysOZM.

V. Authors Bio

Afonso Alemão received a Bachelor of Science degree in Electrical and Computer Engineering from Instituto Superior Técnico, Lisbon, Portugal in 2022. He is currently pursuing a Master of Electrical and Computer Engineering at the Instituto Superior Técnico, Lisbon, Portugal. He is also participating at "JEEC - Jornadas da Engenharia Electrotécnica e de Computadores" as a member of the Web Development Team.

Rui Daniel received a Bachelor of Science degree in Electrical and Computer Engineering from Instituto Superior Técnico, Lisbon, Portugal in 2022. He is currently pursuing a Master of Electrical and Computer Engineering at the Instituto Superior Técnico, Lisbon, Portugal. He is also participating at "JEEC - Jornadas da Engenharia Electrotécnica e de Computadores" as a member of the Web Development Team.



Tomás Fonseca received a Bachelor of Science degree in Electrical and Computer Engineering from Instituto Superior Técnico, Lisbon, Portugal in 2022. He is currently pursuing a Master of Electrical and Computer Engineering at the Instituto Superior Técnico, Lisbon, Portugal.

ATTACHMENTS

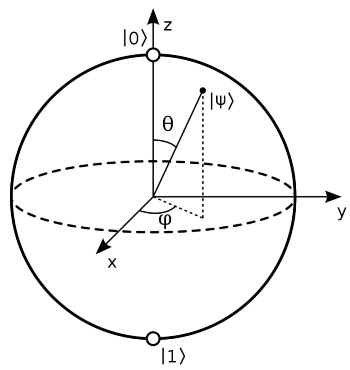


Fig. 1. The Bloch Sphere.

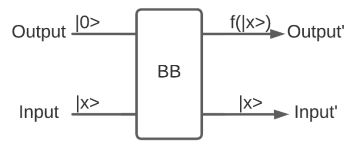


Fig. 2. Write nonreversible functions in a reversible way.

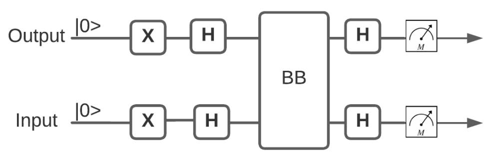


Fig. 3. Circuit to determine whether the function is constant or variable in a quantum computer with a single query.

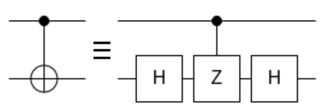


Fig. 4. CNOT gate equivalent circuit.

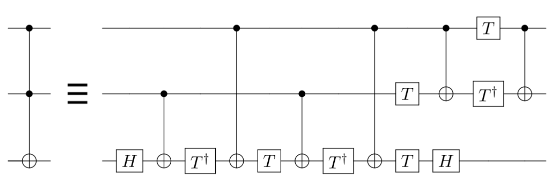


Fig. 5. Toffoli gate equivalent circuit.

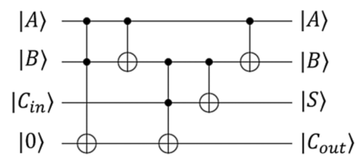


Fig. 6. Circuit model of a quantum full-adder.

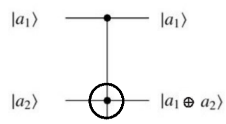


Fig. 7. CNOT gate operation.

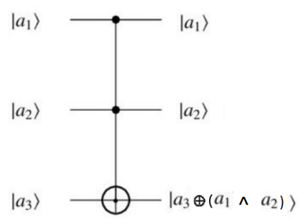


Fig. 8. Toffoli gate operation.

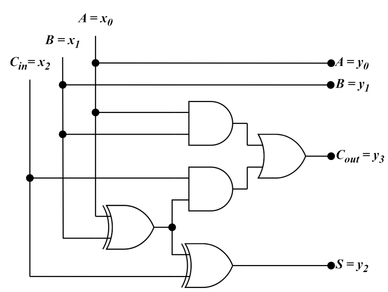


Fig. 9. Logic function diagram of the Quantum Full Adder.

x_3x_2 \ x_1x_0	$\overline{x_1}\,\overline{x_0}$	$\overline{x_1}x_0$	x_1x_0	$x_1\overline{x_0}$
$\overline{x_3}\,\overline{x_2}$	0	1	0	1
$\overline{x_3}x_2$	1	0	1	0
x_3x_2	1	0	1	0
$x_3\overline{x_2}$	0	1	0	1

Fig. 10. Karnaugh map for y_2 output.

x_3x_2 \ x_1x_0	$\overline{x_1}\,\overline{x_0}$	$\overline{x_1}x_0$	x_1x_0	$x_1\overline{x_0}$
$\overline{x_3}\,\overline{x_2}$	0	0	1	0
$\overline{x_3}x_2$	0	1	1	1
x_3x_2	1	0	0	0
$x_3\overline{x_2}$	1	1	0	1

Fig. 11. Karnaugh map for y_3 output.

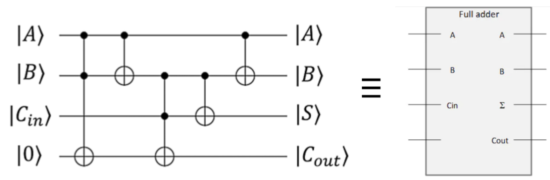


Fig. 12. Two types of implementation of a 2-bit Full Adder.

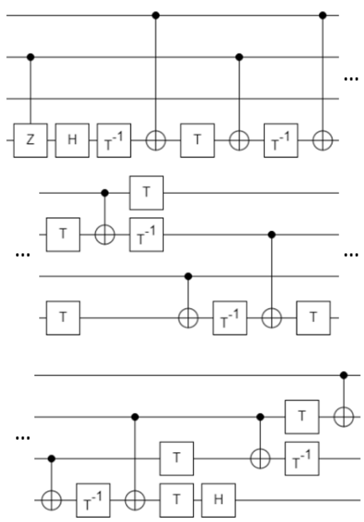


Fig. 13. Quantum Full Adder taking advantage of superposition.

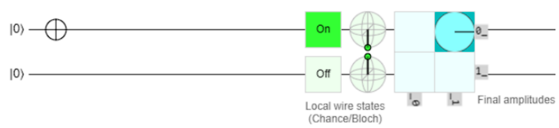


Fig. 14. *Quirk* simulation for Not Gate.

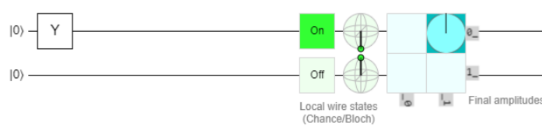


Fig. 15. *Quirk* simulation for Pauli-Y Gate.

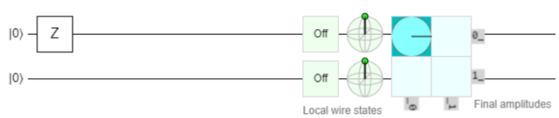


Fig. 16. *Quirk* simulation for Pauli-Z Gate.

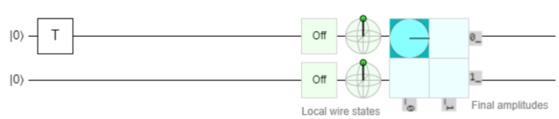


Fig. 17. *Quirk* simulation for T Gate.

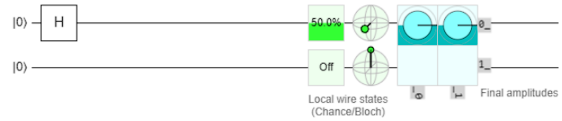


Fig. 18. *Quirk* simulation for Hadamard Gate.

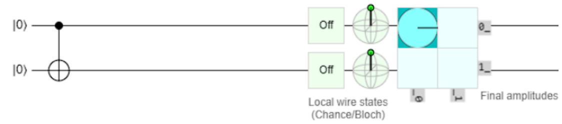


Fig. 19. *Quirk* simulation for CNOT Gate.

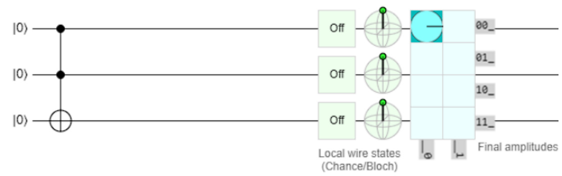


Fig. 20. *Quirk* simulation for Toffoli Gate.

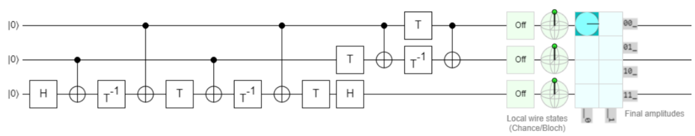


Fig. 21. *Quirk* simulation for Toffoli Gate taking advantage of superposition.

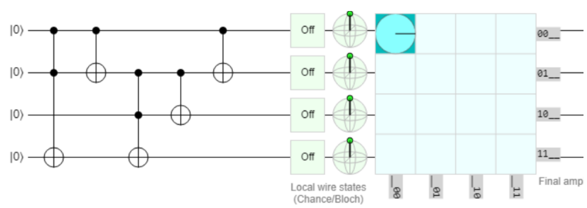


Fig. 22. *Quirk* simulation for Quantum Full Adder.

TABLE I
PAULI-X THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

TABLE II
PAULI-Y THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\begin{pmatrix} 0 \\ i \end{pmatrix}$
$\|1\rangle$	$\begin{pmatrix} -i \\ 0 \end{pmatrix}$

TABLE III
PAULI-Z THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$

TABLE IV
T GATE THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\begin{pmatrix} 0 \\ e^{i\frac{\pi}{4}} \end{pmatrix}$

TABLE V
NOT GATE THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

TABLE VI
HADAMARD GATE THEORETICAL RESULTS.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|+\rangle$
$\|1\rangle$	$\|-\rangle$

TABLE VII
CNOT GATE THEORETICAL RESULTS.

Input		**Output**	
$\|c\rangle$	$\|t\rangle$	$\|y_c\rangle$	$\|y_t\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

TABLE VIII
TOFFOLI GATE THEORETICAL RESULTS.

Input			**Output**		
$\|c_1\rangle$	$\|c_0\rangle$	$\|t\rangle$	$\|y_{c_1}\rangle$	$\|y_{c_0}\rangle$	$\|y_t\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

TABLE IX
QUANTUM FULL ADDER THEORETICAL RESULTS.

Input				Output			
$\|x_3\rangle$	$\|x_2\rangle$	$\|x_1\rangle$	$\|x_0\rangle$	$\|y_3\rangle$	$\|y_2\rangle$	$\|y_1\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$

TABLE X
Quirk SIMULATION RESULTS FOR NOT GATE.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$

TABLE XI
Quirk SIMULATION RESULTS FOR PAULI-Y GATE.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\begin{pmatrix} 0 \\ i \end{pmatrix}$
$\|1\rangle$	$\begin{pmatrix} -i \\ 0 \end{pmatrix}$

TABLE XII
Quirk SIMULATION RESULTS FOR PAULI-Z GATE.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\begin{pmatrix} 0 \\ -1 \end{pmatrix}$

TABLE XIII
Quirk SIMULATION RESULTS FOR T GATE.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\begin{pmatrix} 0 \\ e^{i\frac{\pi}{4}} \end{pmatrix}$

TABLE XIV

Quirk SIMULATION RESULTS FOR HADAMARD GATE.

Input	**Output**
$\|x_0\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|+\rangle$
$\|1\rangle$	$\|-\rangle$

TABLE XV

Quirk SIMULATION RESULTS FOR CNOT GATE.

Input		**Output**	
$\|c\rangle$	$\|t\rangle$	$\|y_c\rangle$	$\|y_t\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

TABLE XVI

Quirk SIMULATION RESULTS FOR TOFFOLI GATE.

Input			**Output**		
$\|c_1\rangle$	$\|c_0\rangle$	$\|t\rangle$	$\|y_{c_1}\rangle$	$\|y_{c_0}\rangle$	$\|y_t\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

TABLE XVII

Quirk SIMULATION RESULTS FOR QUANTUM FULL ADDER.

Input				Output			
$\|x_3\rangle$	$\|x_2\rangle$	$\|x_1\rangle$	$\|x_0\rangle$	$\|y_3\rangle$	$\|y_2\rangle$	$\|y_1\rangle$	$\|y_0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|1\rangle$	$\|0\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$	$\|1\rangle$