Faculdade de Engenharia da Universidade do Porto

Simulador de Circuitos Quânticos

Filipe Daniel Seabra Figueiredo

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof. Dr. Jaime Enrique Villate Matiz

Fevereiro de 2013

U.PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES 2012/2013

A Dissertação intitulada

"Simulador de Computadores Quânticos"

foi aprovada em provas realizadas em 21-03-2013

o júri

Presidente Professor Doutor João Paulo de Castro Canas Ferreira
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Ariel Guerreiro
Professor Auxiliar Departamento de Física da Faculdade de Ciências da Universidade do Porto

Professor Doutor Jaime Enrique Villate Matiz
Professor Auxiliar do Departamento de Engenharia Física da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Filipe Daniel Restier Leal Seabra Figueiredo

Faculdade de Engenharia da Universidade do Porto

Resumo

O desenvolvimento de computadores quânticos, embora seja uma área recente e que ainda apresenta muitos desafios, tem-se mostrado extremamente promissora e de rápido crescimento.

A computação quântica tira proveito de certas propriedades da mecânica quântica para permitir a realização de tarefas de forma bastante mais eficiente do que seria possível na computação clássica. Problemas como a fatorização de números, a procura numa lista não ordenada ou a simulação de sistemas quânticos alcançam resultados significativamente mais rápidos através da computação quântica.

Para tirar partido deste aumento de eficácia é, no entanto, necessário criar algoritmos específicos para os computadores quânticos. A criação de algoritmos quânticos que sejam mais eficientes do que os clássicos é algo de grande dificuldade, uma vez que se tenta tirar proveito das propriedades inerentes à mecânica quântica, como a sobreposição e o entrelaçamento quântico. Assim, a criação de ferramentas que podem auxiliar este processo são de grande importância.

Como tal, o resultado principal deste trabalho é a criação de um programa de simulação de circuitos quânticos que permita ao utilizador criar, editar e simular com facilidade circuitos que traduzem esses algoritmos. Este programa deverá ser útil não só a investigadores da área, como também para fins de ensino.

Abstract

The development of quantum computers, although a recent area which still presents many challenges, has proved to be extremely promising and of rapid growth.

Quantum computation exploits certain properties of quantum mechanics that allow the completion of tasks in a more efficient manner than would be possible with classic computation. Such problems as the number factorization, search of an unstructured list or the simulation of quantum systems achieve results that are significantly faster through quantum computation.

To be able to take advantage of this enhanced performance it is, however, necessary to create specific algorithms for quantum computers. The creation of quantum algorithms that prove to be more efficient than the classical counterparts is a great challenge, since it tries to harness the proprieties inherent to quantum mechanics, such as superposition and quantum entanglement. Thus, the creation of tools that can assist in this process are of great importance.

As such, the main result of this project is to create a quantum circuit simulation program that allows the user to design, edit and easily simulate circuits that translate these algorithms. This program will be useful not only to researchers in the field, as well as for academic purposes.

Índice

Lista de figuras

Lista de tabelas

Abreviaturas e Símbolos

AND	Porta lógica AND (E)
Bits	*Binary Digits* (dígitos binários)
CNOT	Porta quântica controlada NOT (Negada)
NOT	Porta lógica NOT (Negada)
OR	Porta lógica OR (OU)
Qubits	Q*uantum bits* (bits quânticos)

Capítulo 1

Introdução

Neste primeiro capítulo será feita uma breve introdução, apresentando os principais motivos que justificam o estudo do tema e expondo os objetivos a serem atingidos no projeto. Na última secção deste capítulo é ainda feita a descrição da estrutura do documento.

1.1 - Motivação

A invenção dos computadores permitiu à civilização processar informação complexa, de forma eficiente, independente dos cérebros humanos, iniciando assim a era da informação, impulsionando a humanidade para novos horizontes.

Durante dezenas de anos, a evolução da capacidade de processamento dos computadores tem obedecido à Lei de Moore [1]. Esta estipula que o número de transístores que podem ser colocados num dado circuito integrado pelo mesmo preço, duplica a cada dois anos aproximadamente e apresenta como consequência um aumento de performance desses mesmos chips.

No entanto muitos engenheiros e cientistas têm vindo a afirmar que a Lei de Moore pode deixar de se verificar, daqui a uma ou duas décadas, caso não seja inventada uma nova tecnologia [2]. Isto porque depois de reduzir o tamanho dos transístores abaixo de um determinado mínimo, os eletrões podem atravessar o isolante do transístor através de efeitos de túnel quântico, que leva ao escoamento dos electrões, invalidando assim o funcionamento dos mesmos [3].

Assim, torna-se cada vez mais importante encontrar soluções ao problema dos computadores clássicos. Uma das alternativas passa pelos computadores quânticos, que tiram proveito de propriedades da mecânica quântica, como a sobreposição quântica e o entrelaçamento quântico.

A computação quântica é uma área de investigação muito recente e de rápido desenvolvimento. Apesar de existirem atualmente grandes barreiras que deverão ser ultrapassadas para tornar os computadores quânticos viáveis, instituições em vários países já começaram a desenvolver laboratórios de computação quântica.

Deste modo, é importante começarmos a desenvolver em Portugal competências na área da computação quântica. No entanto, devido à multidisciplinaridade do tema, envolvendo teoria da informação, teoria da computação e mecânica quântica, uma metodologia séria da obtenção dessas competências passa por um estudo preliminar teórico bastante extenso.

1.2 – Objetivos

O objetivo principal desta dissertação é a criação e desenvolvimento de um programa de simulação de circuitos quânticos.

Devido à multidisciplinaridade do tema como supra-referido, para adquirir as competências necessárias será efetuado um estudo teórico aprofundado que pode ser dividido essencialmente em três grandes áreas:

- Mecânica quântica.
- Teoria da computação.
- Teoria da informação.

Sendo um simulador de circuitos quânticos uma ferramenta de grande utilidade tanto para investigadores da área como para auxiliar na aprendizagem do tema e uma vez, que as ferramentas já existentes com este propósito não satisfaziam os requisitos procurados, surgiu a oportunidade de criar um programa de simulação de circuitos quânticos.

1.3 – Estrutura

Este documento encontra-se organizado em cinco capítulos.

O primeiro capítulo serve de introdução, apresentando a motivação e os objetivos do projeto.

O segundo capítulo descreve o estado da arte atual, explicitando alguns dos conceitos essenciais para a compreensão deste tema bem como os programas de simulação de circuitos quânticos já existentes.

No terceiro capítulo é feita a especificação dos requisitos considerados para desenvolver o protótipo.

O capítulo quatro expõe o programa de simulação de circuitos quânticos desenvolvido, descrevendo as funcionalidades mais importantes, bem como alguns testes de verificação efetuados.

Finalmente o capítulo cinco apresenta as conclusões do trabalho realizado, assim como as perspetivas de implementações futuras.

Capítulo 2

Estado da arte

Este capítulo apresenta o levantamento do estado da arte, descrevendo inicialmente na secção 2.1, os conceitos teóricos mais importantes para a compreensão do tema. De seguida, em 2.2 são apresentados dois protocolos que reforçam o poder da computação quântica. Na secção 2.3 são descritos alguns dos algoritmos mais importantes conhecidos nos dias de hoje. A secção 2.4 apresenta alguns dos problemas que esses algoritmos podem trazer à criptografia moderna, mas descreve novas soluções de segurança. Finalmente, na secção 2.5, são descritos alguns dos programas atuais de simulação de circuitos quânticos.

2.1 – Conceitos teóricos importantes

Nesta secção serão apresentados alguns dos conceitos teóricos essenciais para a compreensão do tema [4] [5] [6] [7].

2.1.1 – Bits e Qubits

Na computação clássica a informação é representada através de bits, *binary digits* ou dígitos binários. Estes dígitos podem assumir o valor 0 (zero) ou 1 (um). Por exemplo, num computador clássico o bit um pode ser representado por um condensador carregado, e o zero por um descarregado.

Na computação quântica, à semelhança da clássica, a informação é representada por qubits, *quantum bits* ou bits quânticos. Estes qubits podem também assumir o valor de 0 (zero) ou 1 (um), mas contrariamente aos bits clássicos que só podem assumir estados distintos, os qubits podem assumir qualquer sobreposição de zero e um simultaneamente.

Fisicamente os qubits podem ser representados por qualquer sistema quântico que tenha dois estados distintos. Por exemplo, uma das metodologias usadas para a representação de qubits é o *spin* de um eletrão, que assume os dois estados distintos, -1/2 ou +1/2 sendo também habitualmente representado por $\downarrow$ e $\uparrow$. Desta forma, estes dois estados podem-se traduzir no sistema binário zero ou um, permitindo assim a sua utilização num computador quântico.

Os qubits são representados por vetores unitários complexos, utilizando habitualmente a notação de Dirac, em que os dois estados são representados por $|0\rangle$ e $|1\rangle$. Na computação quântica será também útil a notação matricial,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.1}$$

ou no caso geral:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{2.2}$$

onde α e β são dois números complexos com $|\alpha|^2 + |\beta|^2 = 1$, ou seja, para o caso anterior, $|0\rangle$, $\alpha = 1$ e $\beta = 0$. Analogamente, para $|1\rangle$, $\alpha = 0$ e $\beta = 1$.

No caso de termos n qubits, o número de estados possíveis aumenta com 2^n,

$$|\psi\rangle = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{2^{n-1}} \\ \psi_{2^n} \end{bmatrix}. \tag{2.3}$$

2.1.2 – Portas lógicas e quânticas

Para que os bits e qubits sejam úteis na computação, é necessário ter portas capazes de processar essa informação.

Assim na computação clássica foram criadas as portas lógicas como as portas NOT e AND que operam sobre um ou mais bits de entrada, obtendo na saída o estado correspondente à transformação efetuada.

Por exemplo as portas AND, NOT e OR representadas pelos símbolos:

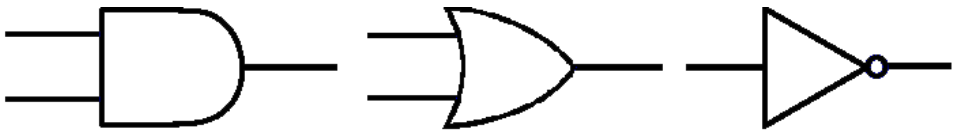

Figura 2.1 - Portas lógicas AND, OR e NOT respetivamente

As portas lógicas anteriores transformam os estados de entrada de acordo com as seguintes tabelas:

Tabela 2.1 - Porta AND

Entrada	Saída
00	0
01	0
10	0
11	1

Tabela 2.2 - Porta OR

Entrada	Saída
00	0
01	1
10	1
11	1

Tabela 2.3 - Porta NOT

Entrada	Saída
0	1
1	0

Da mesma forma, para a computação quântica existem as portas quânticas, estas operam através de matrizes de 2^n linhas por 2^n colunas, onde n representa o número de qubits na entrada do operador. As portas quânticas têm que ser reversíveis, ou seja o número de qubits na saída tem que ser igual ao número de qubits na entrada, ao contrário do que acontece com a porta clássica AND, e têm também que preservar o comprimento de um vetor complexo, como tal as matrizes são também unitárias, ou seja a matriz multiplicada pela própria matriz transposta e conjugada tem que ser igual à identidade, $UU^{\dagger} = \mathrm{I}$.

Por exemplo, uma porta de grande importância, a porta Hadamard é habitualmente representada através do seguinte símbolo:

Figura 2.2 - Porta quântica Hadamard

E a sua matriz corresponde a

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.4}$$

Ou seja quando aplicada ao estado $|0\rangle$:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \tag{2.5}$$

Isto é, o qubit passou a um estado de sobreposição de zero e um.

Aplicando novamente a porta Hadamard ao resultado obtido em (2.5), uma vez que $H = H^{\dagger}$, teremos novamente o estado inicial $|0\rangle$. Da mesma forma se aplicarmos a porta hadamard ao estado $|1\rangle$ obtemos:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \tag{2.6}$$

Outros exemplos de portas que atuam num qubit são as três matrizes de Pauli. A porta Pauli-X é equivalente à porta NOT na computação clássica, ou seja, converte $|0\rangle$ em $|1\rangle$ e $|1\rangle$ em $|0\rangle$. A porta Pauli-Y transforma $|0\rangle$ em $i|1\rangle$ e $|1\rangle$ em $-i|0\rangle$. Finalmente a porta

Pauli-Z não afeta o estado $|0\rangle$, mas transforma $|1\rangle$ em $-|1\rangle$. As suas representações matriciais estão expressas em (2.7)

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.7}$$

Uma outra porta muito usada é a porta CNOT, *controlled NOT gate* ou porta negada controlada representada na seguinte figura:

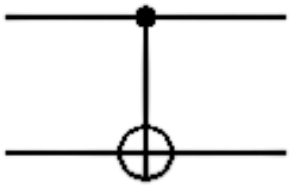

Figura 2.3 - Porta quântica CNOT

Esta porta atua em dois qubits como se pode ver na figura, sendo representada por uma matriz quatro por quatro

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{2.8}$$

A porta não altera o estado do sistema se o primeiro qubit, o qubit de controlo, estiver a $|0\rangle$. Mas nega o segundo qubit se o primeiro estiver a $|1\rangle$. Ou seja, se os qubits na entrada estiverem no estado $|10\rangle$, a porta vai negar o segundo qubit, obtendo na saída o estado $|11\rangle$.

2.1.3 – Medição

Quando é feita uma medição, livre de erros, a um bit na computação clássica, tem-se a certeza do seu resultado e esta não altera o estado do bit. No entanto, na computação quântica, a medição é feita de modo a que o sistema quântico interaja de forma controlada com um sistema externo, obtendo-se, assim, o resultado pretendido. De salientar que esta interação altera o estado final do qubit.

No caso de termos o estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, depois de feita uma medição na base computacional, teremos obtido o resultado $|0\rangle$ com $|\alpha|^2$ de probabilidade e obteremos o resultado $|1\rangle$ com $|\beta|^2$ de probabilidade. O estado final de $|\psi\rangle$ terá colapsado para $|0\rangle$ ou $|1\rangle$ dependendo do resultado obtido. Se não soubermos o estado inicial do sistema, uma vez que o qubit colapsou devido à medição, é impossível determinar os valores de α e β. Ou seja, num sistema de n qubits com 2^n estados diferentes, é possível extrair apenas n bits de informação.

2.1.3.1 – Medição parcial

No caso da medição na base computacional ser feita apenas no primeiro qubit, sendo o sistema de dois qubits, que se representa pela seguinte expressão

$$|\psi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle, \tag{2.9}$$

a probabilidade de obter $|0\rangle$ ou $|1\rangle$ é dada pela soma das probabilidades associadas a esses estados

$$pr(|0\rangle) = pr(|00\rangle) + pr(|01\rangle) = |\alpha_{00}|^2 + |\alpha_{01}|^2 \tag{2.10}$$

$$pr(|1\rangle) = pr(|10\rangle) + pr(|11\rangle) = |\alpha_{10}|^2 + |\alpha_{11}|^2. \tag{2.11}$$

Uma vez que foi efetuada uma medição apenas no primeiro qubit, o estado do segundo qubit terá que ser normalizado para garantir que o vector continua a ter comprimento unitário. No caso de se ter obtido $|0\rangle$ no primeiro qubit, o estado do sistema, após a medição, será

$$|\psi_0\rangle = \frac{\alpha_{00}|00\rangle + \alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2 + |\alpha_{01}|^2}} \tag{2.12}$$

e, analogamente, caso se tenha obtido $|1\rangle$ no primeiro qubit o estado final do sistema será

$$|\psi_1\rangle = \frac{\alpha_{10}|10\rangle + \alpha_{11}|11\rangle}{\sqrt{|\alpha_{10}|^2 + |\alpha_{11}|^2}} \tag{2.13}$$

2.1.3.2 – Medição numa base arbitrária

A medição que é habitualmente usada é feita na base computacional que foi utilizada nos exemplos anteriores, no entanto é possível utilizar qualquer base desde que seja formado por vectores ortonormais entre si, por exemplo

$$|v_1\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad \text{e} \quad |v_2\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{2.14}$$

são vectores ortonormais uma vez que $\langle v_1 | v_2 \rangle = 0$ e por isso podem ser utilizados como uma base na medição.

No caso geral onde se pretende medir um qubit que se encontra no estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ através da base ortonormal:

$$|v_1\rangle = \gamma|0\rangle + \delta|1\rangle \quad \text{e} \quad |v_2\rangle = \delta^*|0\rangle - \gamma^*|1\rangle . \tag{2.15}$$

Assim a probabilidade de obter o estado $|v_1\rangle$ é dada por

$$pr(|v_1\rangle) = |\langle v_1|\psi\rangle|^2 = |\left(\gamma^*\langle 0| + \delta^*\langle 1|\right)\left(\alpha\langle 0| + \beta\langle 1|\right)|^2 =$$
$$= |\gamma^*\alpha + \delta^*\beta|^2 \tag{2.16}$$

da mesma forma, a probabilidade de obter o estado $|v_2\rangle$ é dada por

$$pr(|v_2\rangle) = |\langle v_2|\psi\rangle|^2 =$$
$$= |\delta\,\alpha + \gamma\,\beta|^2 \tag{2.17}$$

2.1.4 – Entrelaçamento quântico

O entrelaçamento quântico é uma propriedade de grande importância para a computação quântica. Estando duas partículas entrelaçadas quanticamente, mesmo que sejam separadas por uma distância arbitrária, continuam sempre a possuir propriedades de correlação. Por exemplo, se possuirmos dois eletrões entrelaçados e se efetuar a medição do *spin* de um deles, o outro eletrão apresentará a mesma orientação do *spin* do primeiro.

Este exemplo pode ser facilmente percebido através de um circuito quântico, se possuirmos dois qubits inicialmente a zero isto é, o estado do sistema inicial é $|\psi\rangle = |00\rangle$ e aplicarmos uma porta hadamard ao primeiro qubit seguido de uma porta cnot como ilustra a figura:

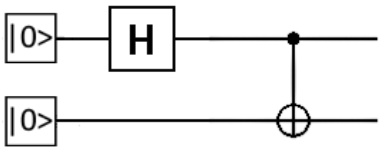

Figura 2.4 - Preparação do estado Bell

depois de aplicada a porta hadamard obteremos

$$|\psi_1\rangle = \frac{|00\rangle + |10\rangle}{\sqrt{2}} \tag{2.18}$$

e de seguida é aplicada a porta CNOT resultando o estado final

$$|\psi_2\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \ . \tag{2.19}$$

Este estado é um dos estados de Bell e encontra-se num estado de entrelaçamento quântico. Mesmo que o primeiro qubit seja separado do segundo, se for efetuada uma medição no primeiro qubit, como se pode verificar pela expressão em (2.19) a probabilidade de obter $|0\rangle$ é 50% e a probabilidade de obter $|1\rangle$ é também 50%. No entanto se tivermos obtido, por exemplo $|0\rangle$ no primeiro qubit, a probabilidade de obtermos $|0\rangle$ no segundo é de 100%. Da mesma forma se tivermos obtido $|1\rangle$ no primeiro, a probabilidade de obter também $|1\rangle$ no segundo é de 100%.

Os quatro estados Bell podem ser obtidos através do circuito da figura 2.4 alterando apenas as entradas para $|01\rangle$, $|10\rangle$ e $|11\rangle$

$$|\phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{2.20}$$

$$|\psi^+\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}} \tag{2.21}$$

$$|\phi^-\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}} \tag{2.22}$$

$$|\psi^-\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}} \tag{2.23}$$

Como foi visto na secção anterior, é possível fazer uma medição numa base arbitrária que não a computacional, desde que sejam escolhidos vectores ortonormais entre si. A medição na base de Bell é de grande importância na computação quântica, podendo-se utilizar por exemplo os vectores $|\phi^+\rangle$ e $|\phi^-\rangle$.

2.2 – Protocolos quânticos

De seguida serão apresentados dois protocolos, a codificação super-densa e a teleportação quântica, com resultados particularmente interessantes, já que não é possível obter um resultado semelhante na computação clássica.

2.2.1 – Codificação super-densa

Anteriormente foi visto que depois de feita uma medição a um sistema de n qubits, apesar de a computação poder atuar em 2^n estados em simultâneo devido à sobreposição, o resultado obtido apenas nos fornece n bits de informação, tal como na computação clássica.

Mas tirando partido de algoritmos específicos e de propriedades exclusivas da computação quântica, como o entrelaçamento quântico, é possível realizar tarefas inacessíveis à computação clássica.

Uma destas tarefas é o protocolo de codificação super-densa ou *superdense coding*, que consiste em enviar do ponto A para um outro ponto B distante, dois bits de informação clássica através de um canal de apenas um qubit [8]. Para esta situação ser possível, é necessário previamente preparar um estado entrelaçado, como o estado $\left|\phi^{+}\right\rangle$ obtido na secção anterior em (2.20), e enviar o primeiro qubit para o ponto A e o segundo para o ponto B. De seguida, no ponto A, dependendo da informação que se pretende enviar para o ponto B, aplica-se uma de quatro sequências de portas ao qubit em posse e, de seguida envia-se o qubit para o ponto B através do canal.

- Se for pretendido enviar os bits 00, aplica-se uma matriz identidade, que é equivalente a não se aplicar nenhuma porta ao qubit. Como de seguida o qubit foi enviado para o ponto B, podemos concluir que o estado em B é na mesma $\left|\phi^{+}\right\rangle$.
- Se for pretendido enviar 01 aplica-se a matriz Pauli-X, ou seja o primeiro qubit é negado, obtendo-se no final no ponto B o estado $\left|\psi^{+}\right\rangle$ (2.21).
- Se for pretendido enviar 10 aplica-se a matriz Pauli-Z, o que se traduz num sinal de menos se o primeiro qubit estiver a $\left|1\right\rangle$, obtendo-se assim o estado $\left|\phi^{-}\right\rangle$ (2.22) no ponto B.
- Se for pretendido enviar 11 aplica-se a matriz Pauli-X seguida da matriz Pauli-Z, ou seja o primeiro qubit é negado, e de seguida fica com um sinal de menos se estiver a $\left|1\right\rangle$, obtendo-se no final em B o estado $\left|\psi^{-}\right\rangle$ (2.23).

No ponto B, agora com um dos estados anteriores, é aplicada uma porta hadamard ao primeiro qubit seguida de uma porta CNOT, sendo feita uma medição no final na base computacional. A figura seguinte ilustra o procedimento:

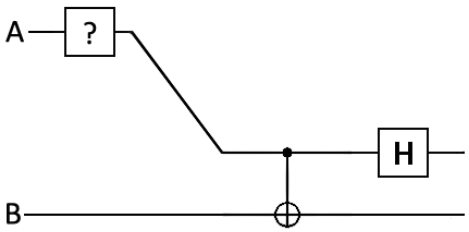

Figura 2.5 - Codificação super-densa

Podemos verificar que para cada um dos quatro casos obteremos expressões:

$$H_1 * \left(CNOT * \left|\phi^{+}\right\rangle\right) = H_1 * \left(CNOT * \frac{\left|00\right\rangle + \left|11\right\rangle}{\sqrt{2}}\right) = H_1 * \frac{\left|00\right\rangle + \left|10\right\rangle}{\sqrt{2}} = \left|00\right\rangle \qquad (2.24)$$

$$H_1 * \left(CNOT * \left|\psi^{+}\right\rangle\right) = H_1 * \left(CNOT * \frac{\left|01\right\rangle + \left|10\right\rangle}{\sqrt{2}}\right) = H_1 * \frac{\left|01\right\rangle + \left|11\right\rangle}{\sqrt{2}} = \left|01\right\rangle \qquad (2.25)$$

$$H_1 * \left(CNOT * \left|\phi^{-}\right\rangle\right) = H_1 * \left(CNOT * \frac{\left|00\right\rangle - \left|11\right\rangle}{\sqrt{2}}\right) = H_1 * \frac{\left|00\right\rangle - \left|10\right\rangle}{\sqrt{2}} = \left|10\right\rangle \qquad (2.26)$$

$$H_1 * \left(CNOT * \left(\left|\psi^-\right\rangle\right)\right) = H_1 * \left(CNOT * \frac{\left|01\right\rangle - \left|10\right\rangle}{\sqrt{2}}\right) = H_1 * \frac{\left|01\right\rangle - \left|11\right\rangle}{\sqrt{2}} = \left|11\right\rangle \qquad (2.27)$$

Finalmente, é feita a medição do estado final. Das expressões anteriores obtêm-se os bits 00, 01, 10 e 11 respetivamente. Assim foi possível obter dois bits de informação através do envio de apenas um qubit, desde que cada um dos pontos tivesse previamente obtido um qubit entrelaçado.

É de notar que se poderia ter utilizado outro estado entrelaçado no início, tal como se poderia utilizar outras matrizes no ponto A, desde que destas resultassem quatro estados distintos ortonormais. No final do procedimento também era possível substituir a porta CNOT e a porta hadamard bem como as medições na base computacionais por uma medição na base de Bell. No entanto o resultado final seria idêntico.

2.2.2 – Teleportação quântica

O protocolo de teleportação quântica foi introduzido em 1993 [9], e consiste em transmitir o estado exato de um qubit para um outro distante, sem deslocar o qubit no espaço físico entre eles, apenas com a transmissão de dois bits clássicos.

Este protocolo, tal como o anterior, tem como base a comunicação prévia de um sistema de dois qubits entrelaçados, como o estado $\left|\phi^+\right\rangle$ obtido em (2.20), onde um é enviado para o ponto A e outro para o ponto B.

O ponto A possui um qubit num estado desconhecido $\left|\psi\right\rangle = \alpha\left|0\right\rangle + \beta\left|1\right\rangle$, para transmitir este qubit para o ponto B através do protocolo de teleportação quântica, só é necessário recorrer a uma medição na base de Bell deste qubit com o qubit que já tinha sido previamente preparado. O estado do sistema imediatamente antes da medição é dado por

$$\left|\psi\right\rangle * \left|\phi^+\right\rangle = \frac{\alpha\left|000\right\rangle + \alpha\left|011\right\rangle + \beta\left|100\right\rangle + \beta\left|111\right\rangle}{\sqrt{2}}, \qquad (2.28)$$

de seguida, para proceder à medição na base de Bell é feita uma mudança de base, da expressão anterior obtendo

$$\frac{1}{2}\left[\left|\phi^+\right\rangle * \left(\alpha\left|0\right\rangle + \beta\left|1\right\rangle\right) + \left|\psi^+\right\rangle * \left(\beta\left|0\right\rangle + \alpha\left|1\right\rangle\right) + \left|\phi^-\right\rangle * \left(\alpha\left|0\right\rangle - \beta\left|1\right\rangle\right) + \left|\psi^-\right\rangle * \left(\beta\left|0\right\rangle - \alpha\left|1\right\rangle\right)\right], \qquad (2.29)$$

da expressão anterior pode-se verificar que a probabilidade de obter cada um dos seguintes estados é 25%:

- Se for obtido $\left|\phi^+\right\rangle$ na medição de base Bell no ponto A, o estado final no ponto B será $\left|\psi\right\rangle = \alpha\left|0\right\rangle + \beta\left|1\right\rangle$.
- Se o resultado no ponto A for $\left|\psi^+\right\rangle$, obtêm-se no ponto B $\left|\psi\right\rangle = \beta\left|0\right\rangle + \alpha\left|1\right\rangle$
- Caso seja obtido $\left|\phi^-\right\rangle$, o estado final no ponto B será $\left|\psi\right\rangle = \alpha\left|0\right\rangle - \beta\left|1\right\rangle$.
- Finalmente, se for obtido $\left|\psi^-\right\rangle$, o qubit em B será $\left|\psi\right\rangle = \beta\left|0\right\rangle - \alpha\left|1\right\rangle$

Desta forma, o ponto A só tem que dizer ao ponto B qual dos quatro resultados é que obteve, que corresponde a dois bits clássicos de informação, através de um meio convencional de comunicação clássica. Com essa informação, no ponto B só é necessário recorrer a uma de quatro portas para recuperar o estado original, dependendo da informação.

- Se a informação transmitida tiver sido $\left|\phi^{+}\right\rangle$, o ponto B aplica a matriz identidade, que equivale a não aplicar nenhuma porta.
- Caso seja obtido $\left|\psi^{+}\right\rangle$, aplica-se no ponto B a matriz Pauli-X.
- Se o resultado tiver sido $\left|\phi^{-}\right\rangle$, o ponto B terá que aplicar uma matriz Pauli-Z.
- Se for obtido o último caso $\left|\psi^{-}\right\rangle$, é aplicado no ponto B a matriz Pauli-Z seguido de Pauli-X.

Obtendo assim em todos os casos anteriores, o estado original $\left|\psi\right\rangle = \alpha\left|0\right\rangle + \beta\left|1\right\rangle$ em B, com a comunicação de apenas dois bits de informação clássica.

É de notar que não houve transmissão física da energia ou matéria, apenas a informação do estado foi transmitida para o ponto B, e esta não foi transmitida mais rápida que a velocidade da luz, uma vez que é necessário enviar a informação através de meios de comunicação clássicos para ter a certeza que o estado é recuperado.

Este protocolo já foi realizado em experiências de laboratório por várias vezes, sendo o recorde atual em termos de distância de uma teleportação quântica 143km, efetuado entre duas ilhas Canárias, La Palma e Tenerife [10].

2.3 – Algoritmos quânticos

Através da computação quântica, tirando proveito de propriedades como a sobreposição e o entrelaçamento, é possível realizar tarefas mais eficientemente do que na computação clássica. No entanto, para tal acontecer, é necessário criar algoritmos quânticos específicos.

De seguida descrevem-se alguns dos algoritmos quânticos mais importantes criados até hoje.

2.3.1 – Algoritmo de Deutsch–Jozsa

O algoritmo de Deutsch-Jozsa proposto em 1992 [11], explica o seguinte problema. Se possuirmos uma função $f:\{0,1\}^{n} \to \{0,1\}$, esta pode ser constante, caso devolva sempre o mesmo valor, ou equilibrada, se devolver zero para metade das entradas e um para a outra metade. Esta função encontra-se dentro de uma caixa preta chamada oráculo, ou seja não sabemos como ela atua, apenas temos acesso às entradas e saídas. Assim, o problema é descobrir se esta função é constante ou equilibrada, observando apenas as entradas e saídas.

Classicamente, no pior cenário, é necessário avaliar o oráculo com mais de metade das combinações possíveis de n, ou seja $2^{n-1}+1$ vezes para ter a certeza que o resultado está correto, assumindo que a função só pode ser constante ou equilibrada.

Utilizando o algoritmo de Deutsch-Jozsa é possível obter o resultado correto com certeza, executando apenas uma avaliação do oráculo, tirando partido da propriedade da sobreposição.

Num caso mais específico deste algoritmo, num sistema onde $n=1$, o algoritmo pode ser representado pelo seguinte circuito:

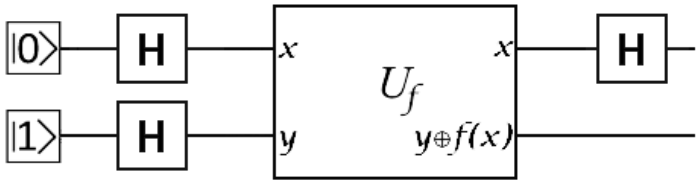

Figura 2.6 - Algoritmo de Deutsch

A função será constante se $f(0)=f(1)$ e será equilibrada se $f(0)\neq f(1)$.

A porta U_f representa uma matriz unitária que transforma a entrada $|x,y\rangle$ em $|x,y\oplus f(x)\rangle$, onde $\oplus$ é a soma em módulo dois.

O estado do sistema antes da porta U_f é dado por

$$|\psi_1\rangle=\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right).\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right). \tag{2.30}$$

Como a porta U_f transforma o estado

$$|x\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}\xrightarrow{U_f}(-1)^{f(x)}|x\rangle.\frac{|0\rangle-|1\rangle}{\sqrt{2}} \tag{2.31}$$

Podemos concluir que transformará $|\psi_1\rangle$ em

$$|\psi_2\rangle=\begin{cases}\pm\left(\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right).\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) & se\ f(0)=f(1)\\ \pm\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right).\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) & se\ f(0)\neq f(1)\end{cases} \tag{2.32}$$

Finalmente, depois de aplicada a porta hadamard ao primeiro qubit o estado final do sistema é dado por

$$|\psi_3\rangle=\begin{cases}\pm|0\rangle.\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) & se\ f(0)=f(1)\\ \pm|1\rangle.\left(\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right) & se\ f(0)\neq f(1)\end{cases} \tag{2.33}$$

A equação (2.33) também pode ser escrita de forma mais compacta:

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \tag{2.34}$$

Desta forma pode-se verificar da equação (2.33) ou (2.34) que se a função for constante, depois de feita uma medição na base computacional do primeiro qubit, obtêm-se zero. Se a função for equilibrada, então obtêm-se o resultado um depois de feita a medição.

O algoritmo mais geral, Deutsch-Jozsa, funciona essencialmente da mesma forma, mas aplicado a n qubits. Ou seja, o sistema será de $n+1$ qubits tal como anteriormente.

O procedimento passa por utilizar a porta hadamard aplicada aos primeiros n qubits, inicialmente no estado $|0\rangle$ e o qubit da posição $n+1$, inicialmente no estado $|1\rangle$ passará também por uma porta hadamard. De seguida é utilizada a porta U_f com $n+1$ entradas ao sistema e, aplica-se novamente a porta hadamard aos primeiros n qubits.

Finalmente é feita uma medição nos primeiros n qubits na base computacional como anteriormente, e se for obtido apenas zeros, então a função é constante, caso contrário a função é equilibrada.

2.3.2 – Algoritmo de Shor

O algoritmo de Shor [12] publicado em 1994 é o algoritmo mais famoso dos dias de hoje devido à sua capacidade de factorizar números de forma eficiente, em tempo polinomial, ao contrário do melhor algoritmo clássico conhecido, já que este é considerado ter complexidade sub-exponencial.

O algoritmo para factorizar o número inteiro N baseia-se em encontrar o período da função $F(x) = a^x \bmod N$. O valor de a é escolhido aleatoriamente de forma que $a < N$. De seguida, encontra-se o período r da função $F(x)$, este depende dos valores de a e N. Se r for um número ímpar ou se $a^{r/2} = -1$ então começasse o procedimento do início com um valor diferente para a. Caso contrário determina-se o máximo divisor comum $\gcd(a^{r/2} \pm 1, N)$, sendo os valores obtidos o resultado da factorização de N.

Este algoritmo pode ser executado classicamente, no entanto encontrar o período r continua a ser um problema que evolui exponencialmente com N. Mas utilizando a transformada de Fourier quântica, que pode ser implementada eficientemente na computação quântica, é possível encontrar o período r em tempo polinomial e consequentemente factorizar um número N de forma eficiente.

2.3.3 – Algoritmo de Grover

O algoritmo Grover, proposto em 1996 [13], é habitualmente referenciado como sendo um algoritmo de procura de base de dados ou listas não ordenados, no entanto não se limita apenas a este objetivo. Pode também ser aplicado à maior parte dos algoritmos clássicos que utilizam algum tipo de procura, conseguindo assim um aumento de eficiência quadrática $O\left(\sqrt{N}\right)$.

Este algoritmo é probabilístico, isto é, determina o resultado correto com uma probabilidade alta de sucesso, podendo esta probabilidade ser melhorada com sucessivas iterações.

A figura seguinte exemplifica o circuito quântico do algoritmo:

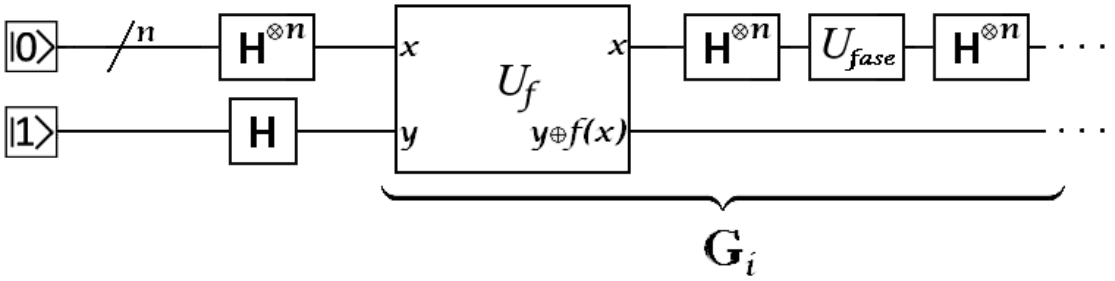

Figura 2.7 - Algoritmo de Grover

O símbolo $/^n$ representa um conjunto de n qubits e a porta $H^{\otimes n}$ é equivalente a utilizar a porta hadamard em cada um dos n qubits.

A porta U_f representa uma matriz unitária que transforma a entrada $|x, y\rangle$ em $|x, y \oplus f(x)\rangle$, onde $\oplus$ é a soma em módulo dois, tal como foi visto anteriormente no algoritmo de Deutsch-Jozsa.

A função $f(x)$ é construída de forma que $f(x)=1$ se x for solução do problema e $f(x)=0$ se x não for solução do problema.

Finalmente, a porta U_{fase} faz uma mudança de fase ao sistema, isto é, reverte o sinal de todos os termos de $|x\rangle$ exceto para $|0\rangle$.

O conjunto de portas representado por G_i é repetido $O\left(\sqrt{N}\right)$ vezes. No final do circuito os primeiros n qubits são medidos na base computacional para obter o resultado correto com uma probabilidade elevada.

O algoritmo clássico equivalente tem complexidade $O(N)$, enquanto que o algoritmo de Grover apenas $O\left(\sqrt{N}\right)$. O que se traduz num aumento quadrático de eficiência que embora não seja tão significativo como no algoritmo de Shor, para N elevados já se torna bastante relevante.

2.4 – Criptografia

A criptografia é uma área de extrema importância, principalmente nos dias de hoje com a utilização da internet, já que é necessário manter privadas e seguras informações como dados pessoais, números de cartões de crédito, entre outros dados confidenciais.

Serão apresentados de seguida os problemas futuros, induzidos pela computação quântica, na maior parte da criptografia utilizada atualmente mas também, as possíveis soluções obtidas através dessa mesma área, a computação quântica.

2.4.1 – Criptografia clássica

A maioria da criptografia usada nos dias de hoje para comunicar com segurança e privacidade na internet é baseada no algoritmo RSA [14]. Este algoritmo baseia-se no sistema de palavra-chave pública, isto é, são criadas duas palavras-chave, uma privada e outra pública. A palavra-chave pública pode ser usada por qualquer pessoa e serve para encriptar a mensagem, enquanto que a palavra-chave privada é mantida em segurança e pode ser usada para desencriptar o código recebido.

Estas palavras-chave são criadas a partir da multiplicação de dois números primos para criar um número de grandes dimensões (códigos criados hoje em dia têm tipicamente mais de 300 algarismos decimais), o número obtido juntamente com um valor auxiliar compõem a palavra-chave pública. A palavra-chave privada por sua vez pode ser facialmente obtida com o conhecimento dos dois números primos iniciais.

Este algoritmo é considerado seguro uma vez que classicamente, é extremamente difícil encontrar os fatores de números grandes, já que a complexidade do melhor algoritmo clássico encontrado até hoje é sub-exponencial.

No entanto, como foi referido anteriormente, o algoritmo de Shor é capaz de fatorizar números em tempo polinomial num computador quântico. Isto significa que, quando existirem computadores quânticos com o número suficiente de qubits, será possível quebrar qualquer código de criptografia baseado na fatorização de números como o RSA.

Esta conclusão simultaneamente demonstra a eficiência do algoritmo quântico de Shor face ao algoritmo clássico, como levanta um problema de privacidade e segurança na criptografia usada maioritariamente nos dias de hoje.

2.4.2 – Criptografia quântica

Apesar do problema de segurança introduzido na criptografia atual pela computação quântica, existem várias soluções, sendo que uma delas passa por uma nova área, a criptografia quântica, mais concretamente a distribuição de palavras-chave quânticas.

Esta solução baseia-se no sistema de palavras-chave privadas, isto é, as entidades que pretendem comunicar com segurança e privacidade, escolhem uma palavra-chave previamente e partilham-na entre eles, através de uma via que consideram segura. Com esta

palavra-chave podem encriptar as mensagens e envia-las com segurança através de qualquer meio de comunicação, uma vez que só é possível desencriptar a mensagem com o conhecimento da palavra-chave correta. O problema deste sistema é que se uma terceira entidade estiver a espiar o canal de comunicação, no momento da partilha da palavra-chave, poderá então desencriptar qualquer mensagem partilhada comprometendo a segurança das mesmas.

O sistema de distribuição de palavras-chave quânticas, tira partido das propriedades da mecânica quântica, para partilhar palavras-chave com segurança. Um dos protocolos que permite esta partilha, proposto em 1984 com o nome de BB84 [15], pode ser explicado da seguinte forma.

O emissor gera um bit clássico aleatoriamente e de seguida, escolhe também aleatoriamente, entre duas bases diferentes, para criar um qubit num dos quatro estados seguintes:

Tabela 2.4 - Preparação do qubit para enviar no protocolo BB84

Base	**Bit**	
	0	1
X	$\psi_{X0} = \|0\rangle$	$\psi_{X1} = \|1\rangle$
Y	$\psi_{Y0} = \frac{\|0\rangle + \|1\rangle}{\sqrt{2}}$	$\psi_{Y1} = \frac{\|0\rangle - \|1\rangle}{\sqrt{2}}$

Como se pode verificar, os quatro estados não são ortogonais e como tal, não é possível medir os quatro estados com certeza utilizando sempre a mesma base.

Este procedimento é repetido para n qubits e de seguida, os qubits são enviados para o recetor. Uma vez que o emissor ainda não revelou em que base é que preparou os qubits, o recetor escolhe aleatoriamente uma das duas bases para medir cada qubit, registando os valores 0 ou 1 obtidos.

No final de concluir as medições, o recetor utiliza um canal de comunicação clássico público, para avisar o emissor que concluiu as medições. Agora, tanto o recetor como o emissor anunciam que base utilizaram para cada qubit, sem comunicar o valor obtido. Desta forma, depois de descartar os valores medidos em bases diferentes e guardar apenas os valores correspondentes à mesma base, podem ter a certeza que possuem a mesma sequência.

Caso uma terceira entidade estivesse a espiar o canal de comunicação, uma vez que não tinha conhecimento das bases originais, a medição feita vai inevitavelmente colapsar o estado do qubit, introduzindo erros na comunicação.

Assim, o recetor e o emissor partilham apenas uma parte da sequência obtida num canal de comunicação público, para verificar a veracidade do código. Se o código estiver correto, então utilizam a outra parte da sequência para criar a palavra-chave, caso contrário, significa que foi detetada a presença de uma terceira entidade, neste caso repete-se o procedimento no mesmo ou noutro canal, até certificarem a privacidade do mesmo.

Desta forma, apesar de não poderem evitar que uma terceira entidade espie o canal, podem sempre detetar a sua presença, enviando apenas a mensagem depois de assegurarem a segurança da palavra-chave.

2.5 – Programas de simulação de circuitos quânticos

Como foi visto na secção 2.3, a criação de algoritmos quânticos é essencial para tirar partido do aumento da eficiência que a computação quântica pode proporcionar. Assim, o desenvolvimento de ferramentas que auxiliem esta tarefa é de grande utilidade.

De seguida são apresentados alguns dos programas já existentes que podem simular circuitos quânticos.

2.5.1 – libquantum

Libquantum é uma biblioteca para a linguagem C criado na Alemanha [16], que pode ser utilizado para simular circuitos e algoritmos quânticos. É eficiente e possui alguns algoritmos já pré-implementados. No entanto é executado na linha de comandos e é necessário saber programar para o utilizar. Assim sendo, a utilização desta biblioteca para simular circuitos quânticos torna-se bastante complexa.

Na seguinte figura está implementado um circuito e a probabilidade da respetiva medição:

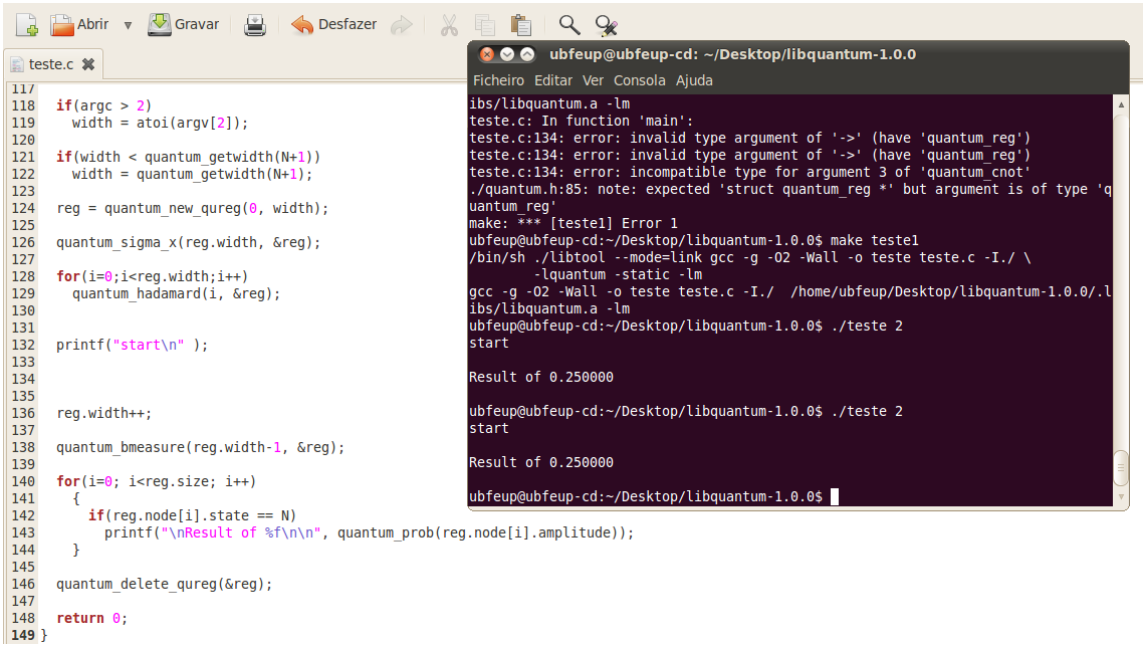

Figura 2.8 - Exemplo de utilização da biblioteca Libquantum

2.5.2 – Simulador de Wire Communications Laboratory

Este programa, criado na Universidade de Patras da Grécia [17], pode ser utilizado em qualquer sistema operativo, uma vez que é executado a partir do servidor do site. No entanto é de difícil utilização, já que não possui uma interface gráfica avançada, e oferece um leque pequeno de portas quânticas para utilizar.

A figura seguinte mostra a simulação de um circuito e os seus resultados:

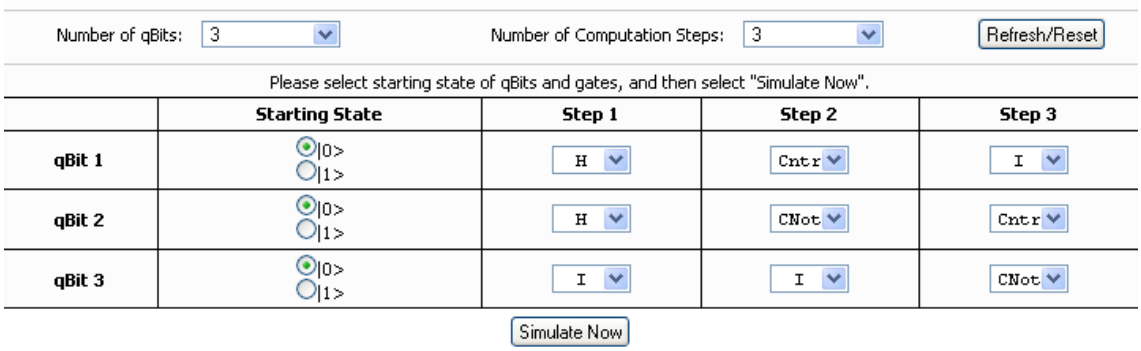

Figura 2.9 - Circuito do simulador de circuitos quânticos de Wire Communications Laboratory

Measure				
	Starting State	**Step 1**	**Step 2**	**Step 3**
qBit State 1	1.00	0.25	0.25	0.25
qBit State 2	0.00	0.00	0.00	0.00
qBit State 3	0.00	0.25	0.25	0.00
qBit State 4	0.00	0.00	0.00	0.25
qBit State 5	0.00	0.25	0.25	0.25
qBit State 6	0.00	0.00	0.00	0.00
qBit State 7	0.00	0.25	0.25	0.00
qBit State 8	0.00	0.00	0.00	0.25
	Starting State	**Step 1**	**Step 2**	**Step 3**
Phase				
	Starting State	**Step 1**	**Step 2**	**Step 3**
qBit State 1	0.00°	0.00°	0.00°	0.00°
qBit State 2	0.00°	0.00°	0.00°	0.00°
qBit State 3	0.00°	0.00°	0.00°	0.00°
qBit State 4	0.00°	0.00°	0.00°	0.00°
qBit State 5	0.00°	0.00°	0.00°	0.00°
qBit State 6	0.00°	0.00°	0.00°	0.00°
qBit State 7	0.00°	0.00°	0.00°	0.00°
qBit State 8	0.00°	0.00°	0.00°	0.00°
	Starting State	**Step 1**	**Step 2**	**Step 3**

Figura 2.10 - Resultados do simulador de circuitos quânticos de Wire Communications Laboratory

2.5.3 – QCAD

O programa QCAD foi criado nas universidades de Tokyo e de Nagoya no Japão [18]. É uma aplicação utilizável apenas no sistema operativo Windows, no entanto tem uma boa interface gráfica e várias portas quânticas. Porém, não é possível visualizar os resultados na forma matricial e não tem opção de criar portas personalizadas pelo utilizador.

A figura seguinte apresenta um circuito e os seus resultados simulados:

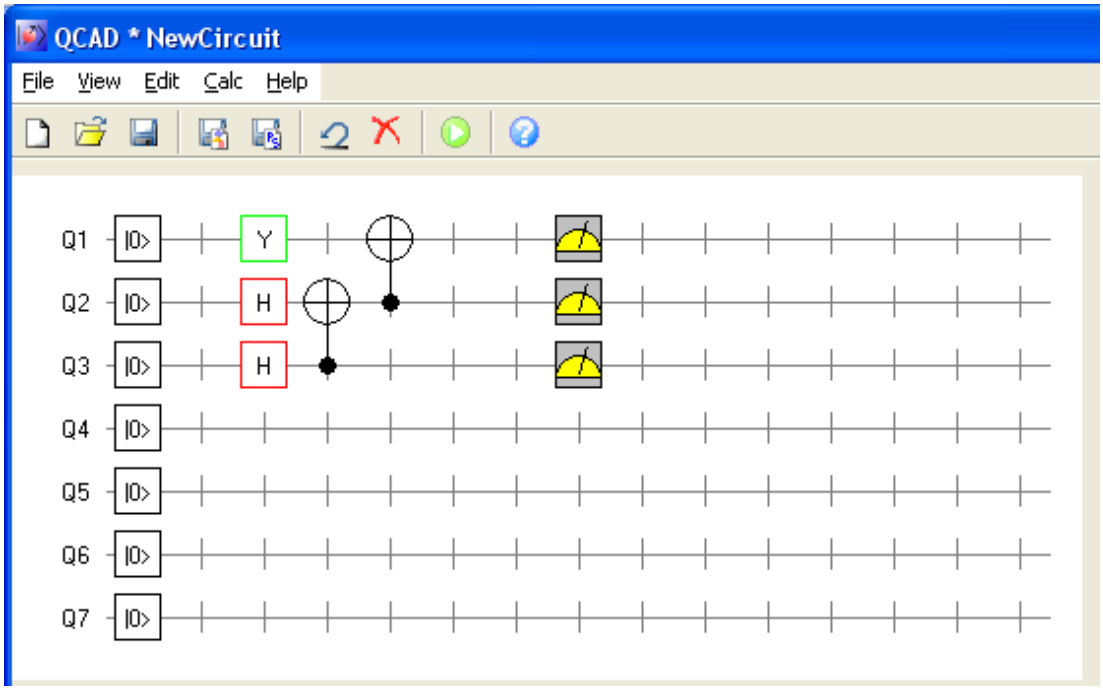

Figura 2.11 - Representação do circuito do programa QCAD

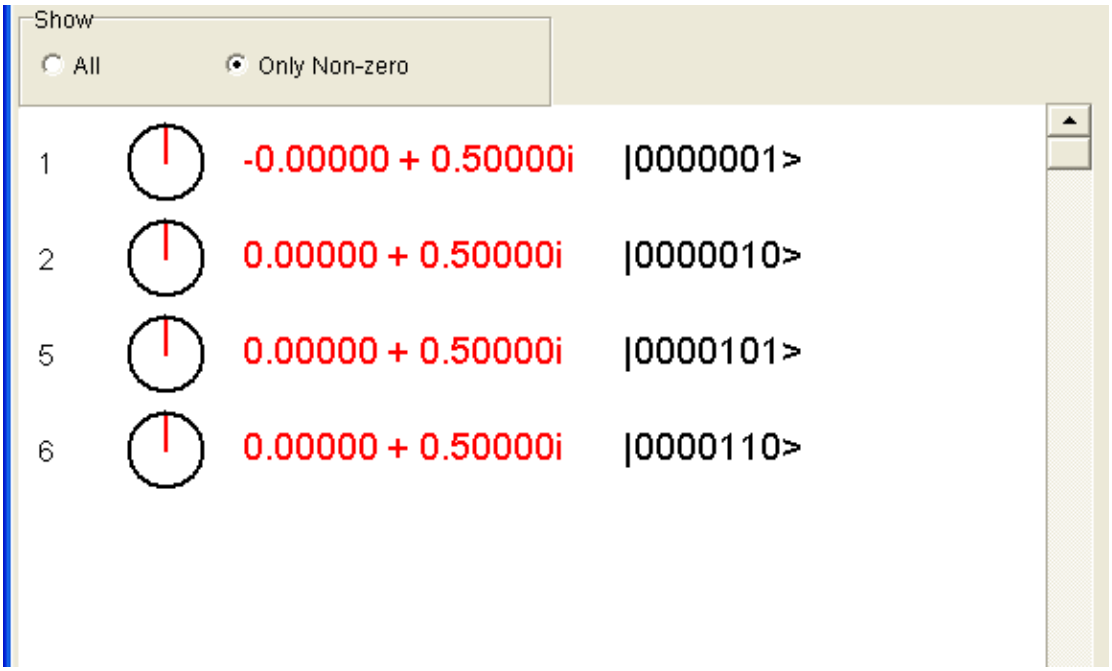

Figura 2.12 - Resultados da simulação do programa QCAD

2.5.4 – jQuantum

O simulador jQuantum foi criado na linguagem de programação java [19], desta forma pode ser utilizado em qualquer sistema operativo desde que este já tenha instalado o *plug-in* necessário. Este programa, tal como o QCAD, possui uma boa interface gráfica e é dos melhores programas atuais para simulação de circuitos quânticos, tem uma grande variedade de portas quânticas predefinidas para utilizar e é eficiente na simulação. No entanto, não possui a opção de adicionar portas personalizadas, e é difícil visualizar os resultados, especialmente quando estes têm diferentes fases.

A figura seguinte apresenta um circuito e os seus resultados simulados:

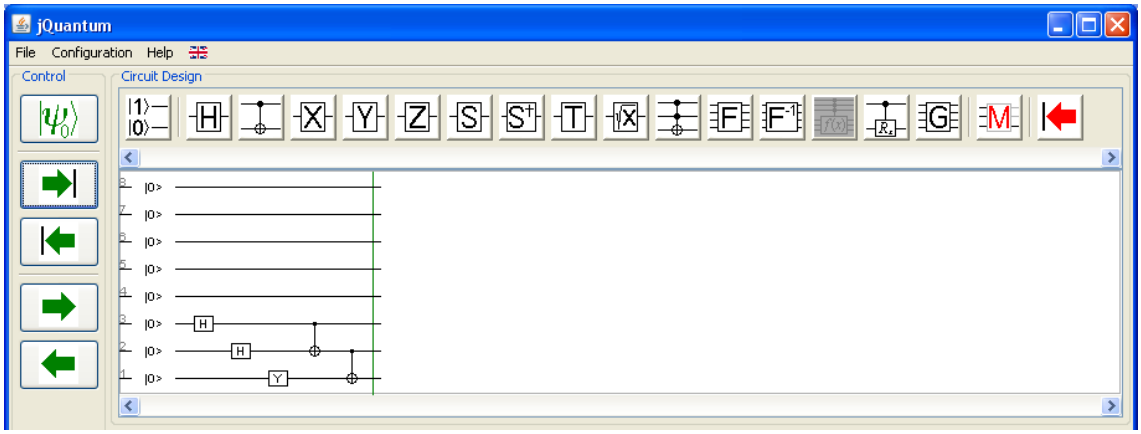

Figura 2.13 - Representação do circuito do programa jQuantum

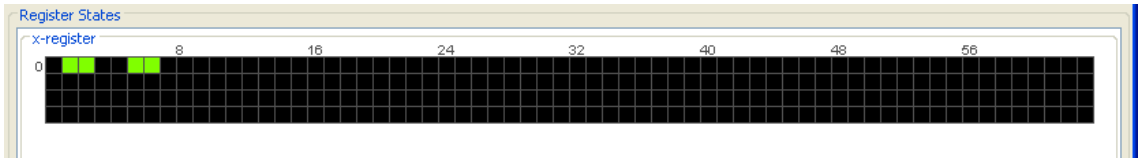

Figura 2.14 - Resultados da simulação do programa jQuantum

Capítulo 3

Especificação

A criação de um programa de simulação de circuitos quânticos, tem como vista simplificar e auxiliar o estudo e a investigação, bem como ajudar a compreensão destes mesmos circuitos e dos algoritmos que representam. Como tal, de modo a assegurar a criação de um protótipo que seja uma ferramenta útil para estes fins, devem ser definidos os requisitos a serem cumpridos no desenvolvimento da aplicação.

Deste modo os requisitos foram divididos em dois grupos, requisitos funcionais, as opções e funções acessíveis ao utilizador, e os requisitos não funcionais, restrições de usabilidade da aplicação.

3.1 – Requisitos funcionais

- O programa deverá oferecer ao utilizador a possibilidade de adicionar, de forma simples, entradas, portas e medições.

- O utilizador deverá poder selecionar facilmente qualquer objeto: portas, entradas e medições, colocadas previamente no ecrã de circuitos.

- O utilizador deverá poder alterar o valor da entrada de um qubit, entre o estado zero e um, tanto no momento de adição do objeto, como posteriormente num momento de edição.

- Os objetos colocados deverão poder ser movidos e reposicionados em qualquer altura pelo utilizador.

- O utilizador deve ter a opção de apagar os objetos selecionados.

- O utilizador deverá poder adicionar portas personalizadas, de tamanho arbitrário, em qualquer ponto do circuito de simulação.

- Devem ser adicionadas opções de gravar (*Save*) e de abrir (*Load*), de modo que o utilizador possa continuar um circuito prévio, ou de partilhar circuitos com outros utilizadores.

- Os resultados da simulação deverão ser de fácil visualização e devem ser apresentados na notação de Dirac e na forma matricial.

3.2 – Requisitos não funcionais

- Deve ser um programa que possa ser executado a partir de qualquer um dos principais sistemas operativos (Windows, Mac OS X, Unix).

- A implementação será feita em flash, na linguagem actionscript, uma vez que mais de 95% [20] dos computadores com ligação à internet possuem a capacidade de executar programas em flash.

- Deverá ter uma interface gráfica, de fácil usabilidade, onde o utilizador poderá construir e visualizar o seu circuito.

- O programa deverá oferecer algumas das portas mais usadas pré-definidamente como ferramentas.

- A aplicação deverá interpretar o circuito corretamente independentemente da posição ou ordem em que o utilizador adicionou ao ecrã as entradas, portas e medições.

- O resultado gerado para cada medição deve considerar corretamente as probabilidades do qubit, assim como a possibilidade de existir entrelaçamento quântico.

- O programa deverá suportar a utilização de números complexos em todas as matrizes e estados do sistema, assim como em todas as funções matemáticas implementadas.

- A aplicação deve ser capaz de interpretar a notação introduzida pelo utilizador na adição de matrizes personalizadas.

- A adição de matrizes definidas pelo utilizador deve ser acompanhada de uma verificação para garantir que estas são do tamanho certo e unitárias, de modo a perfazerem os requisitos de uma porta quântica.

- A aplicação deverá suportar a adição de um número ilimitado de matrizes personalizadas, permitindo o utilizador apagar posteriormente, qualquer uma das matrizes.

- As opções de gravar (*Save*) e de abrir (*Load*), devem possuir um sistema de verificação, para garantir que não existem erros nem ficheiros corrompidos.

- A aplicação deve ser capaz de reconhecer problemas na estrutura do circuito criado pelo utilizador, como o caso de realimentações.

Capítulo 4

Implementação

Este capítulo apresenta inicialmente a interface e as funcionalidades do programa de simulação de circuitos quânticos desenvolvido. Por fim, são feitos alguns testes de verificação do programa.

O programa encontra-se todo na língua Inglesa, com o objetivo de ser uma aplicação mais abrangente, para que possa ser utilizada em qualquer país do mundo.

4.1 – Acessibilidade do programa

O programa foi concebido de forma a poder ser acedido em qualquer computador, independentemente do sistema operativo, e ser de fácil acesso.

Para utilizar a aplicação só é necessário aceder à página web (http://quark.fe.up.pt/qcs) onde se encontra a aplicação, inicializando assim o programa automaticamente, ou descarregar a aplicação para o computador, podendo assim ser utilizado posteriormente sem a necessidade de ter acesso à internet.

4.2 – Janela de interface principal

A interface gráfica dá a possibilidade do utilizador poder facilmente visualizar e editar os circuitos criados para posteriormente simulá-los.

A figura seguinte demonstra a interface principal do programa:

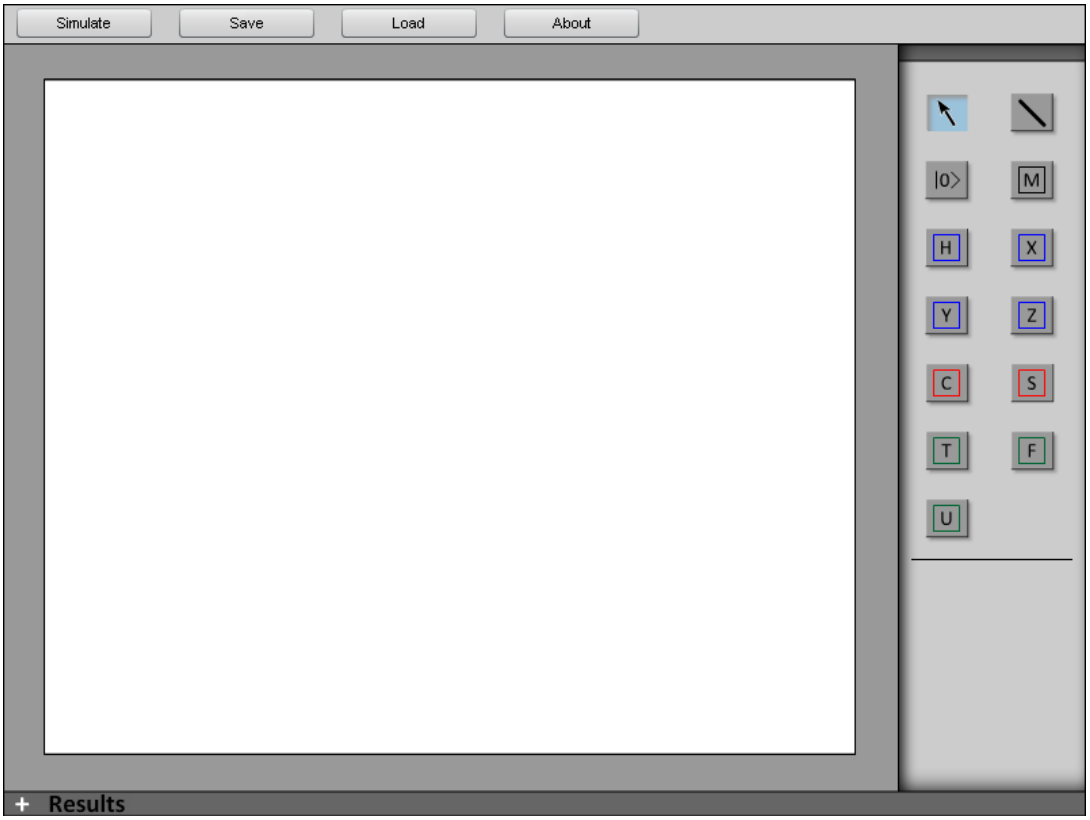

Figura 4.1 - Imagem principal do protótipo criado

Como se pode verificar, do lado direito foi criada uma barra de ferramentas, onde o utilizador pode escolher uma das ferramentas disponíveis, como entradas de qubits ou portas quânticas para criar o seu circuito.

Na barra de cima encontram-se quatro botões que oferecem ao utilizador funcionalidades específicas:

- **Simulate:** Permite ao utilizador simular o circuito construído, apresentando uma nova janela com os resultados no final da simulação.
- **Save:** Oferece a possibilidade de gravar os circuitos criados.
- **Load:** Permite abrir um circuito que tenha sido previamente gravado.
- **About:** Mostra o nome e a versão da aplicação atuais, bem como o nome do autor e disponibiliza um endereço eletrónico para qualquer eventualidade relacionada com o programa, como a deteção de possíveis erros ou sugestões.

Caso o utilizador carregue na barra situada em baixo, uma nova janela irá apresentar os resultados da última simulação realizada.

4.3 – Barra lateral de ferramentas

A barra lateral oferece várias funcionalidades para a criação de circuitos ao utilizador.

Para criar um circuito é necessário primeiro adicionar o número desejado de qubits, isto pode ser realizado através da ferramenta de entrada "*Input Tool*".

De seguida o utilizador deverá adicionar as portas quânticas desejadas através de uma das ferramentas de criação de portas predefinidas, como a Hadamard ou Pauli-X.

Poderá ainda utilizar a ferramenta personalizada "*Custom Tool*" para adicionar matrizes criadas pelo utilizador, desde que estas sejam unitárias.

De modo a ligar as entradas e portas de forma a criar um circuito, será necessário utilizar a ferramenta linha "*Line Tool*". O utilizador deverá carregar no conector de uma das portas que pretende ligar e apenas largar o botão do rato quando se encontrar por cima da segunda porta, criando assim uma linha que ligará as portas.

Se o utilizador pretender adicionar uma medição no final do circuito poderá faze-lo com a ferramenta de medição "*Measurement Tool*".

Com a primeira ferramenta, a seta, o utilizador poderá selecionar, mover e apagar, portas, entradas, medições e linhas previamente adicionadas. Permitirá ainda ver funcionalidades extra específicas a alguns dos itens adicionados ao ecrã, como por exemplo as entradas.

Em circuitos de grandes dimensões, pode ser útil ver as linhas de várias cores diferentes para facilitar a distinção. Assim, quando uma entrada de qubit se encontra selecionada, é possível escolher uma nova cor nas opções adicionais, alterando automaticamente a cor de todas as linhas representadas por esse qubit.

De seguida encontra-se um exemplo de um circuito criado no protótipo com as opções adicionais da entrada de qubits visíveis:

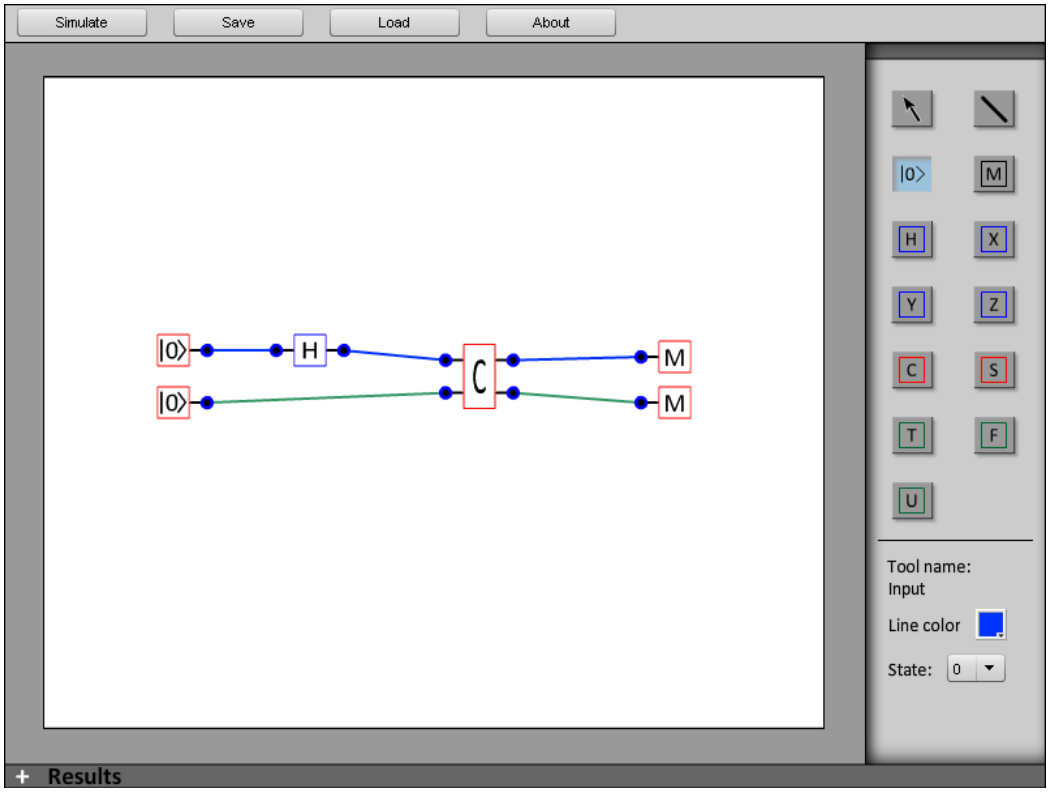

Figura 4.2 - Exemplo de circuito criado

4.4 – Ferramenta de portas personalizadas

É possível adicionar portas personalizadas recorrendo à ferramenta "*Custom*" representada pelo símbolo "U". O utilizador deverá carregar na opção "*Add*" para adicionar uma matriz à lista para poder ser posteriormente usada. Esta ação irá abrir uma nova janela onde será possível introduzir os valores da matriz bem como o seu nome. Finalmente, para guardar a matriz a opção de "*Save*" deverá ser premida. Caso a sintaxe da matriz introduzida esteja correta e seja unitária, esta será adicionada à lista e poderá ser visualizada na forma matricial na área de texto do lado direito.

Para a sintaxe ser corretamente reconhecida, cada elemento da coluna da matriz deverá ser serparada por uma vírgula "," e cada linha por um ponto e vírgula ";". No caso de serem utilizados números complexos a letra "i" deverá ser acompanhar o valor para representar a parte imaginária. Por fim, é ainda possível introduzir raízes quadradas, para tal deverá ser utilizada a sintaxe: "sqrt(x)" onde x representa o valor.

Por exemplo, se o utilizador pretender adicionar a seguinte matriz unitária:

$$M = \begin{bmatrix} 0.5 & 0 & \sqrt{0.75} & 0 \\ \sqrt{0.75} & 0 & -0.5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

Deverá utilizar a seguinte sintaxe:

0.5, 0, sqrt(0.75), 0;
sqrt(0.75), 0, -0.5, 0;
0, 1, 0, 0;
0, 0, 0, 1

Na figura seguinte é possível ver as três primeiras linhas do exemplo anterior:

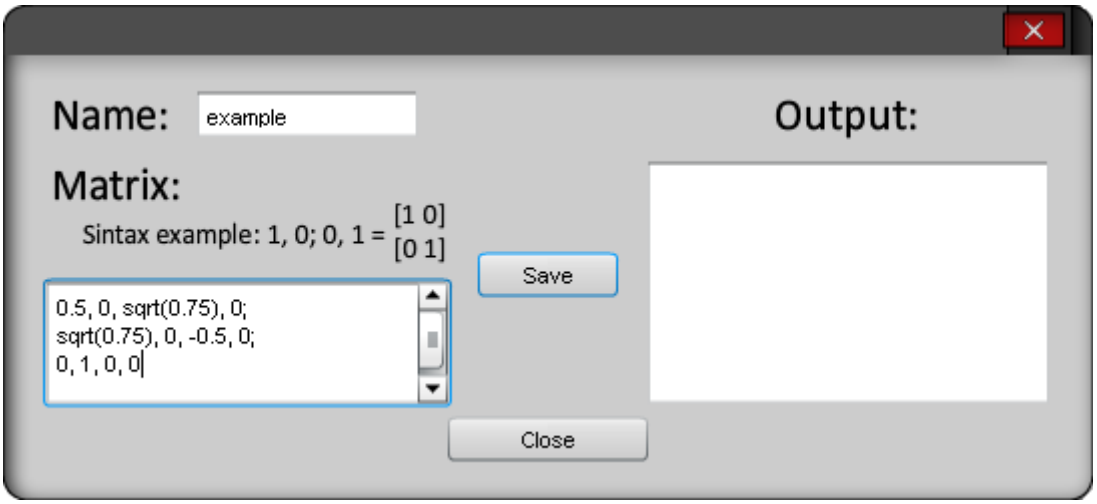

Figura 4.3 - Exemplo de matriz personalizada

4.5 – Simulação

Depois de criado um circuito, poderá ser feita uma simulação recorrendo ao botão "*Simulate*" na barra superior do programa. No final da simulação a aplicação apresentará uma nova janela com os resultados obtidos do circuito.

Os resultados podem ser vistos na notação Dirac ou na forma matricial. O utilizador pode escolher em ver os resultados de acordo com as portas de saída, de cima para baixo, ou de acordo com as entradas.

Caso a aplicação detete algum problema no circuito, como algumas portas estarem ligadas incorretamente, será apresentada uma notificação a avisar o utilizador dessa ocorrência.

Por fim, o utilizador pode ainda alterar o tamanho de letra com a funcionalidade "*Text Size*".

Na figura seguinte encontra-se explicitado um exemplo dos resultados de uma simulação na notação Dirac:

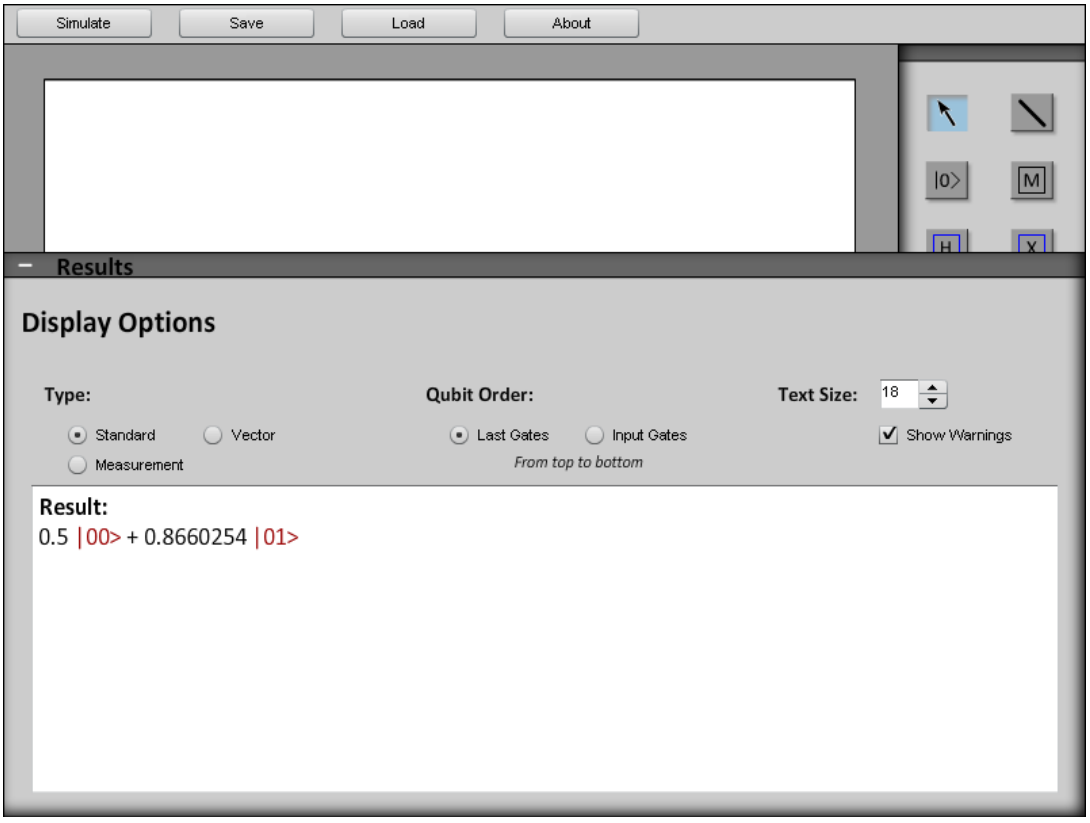

Figura 4.4 - Exemplo da janela de resultados

Se o circuito incluir medições no final, os resultados destas poderão ainda ser visualizados com a opção "*Measurement*". A probabilidade de obter zero ou um em cada uma das medições feitas será apresentada, bem como o resultado obtido na respetiva medição. Este resultado obtido será gerado aleatoriamente pelo programa em cada simulação, respeitando as probabilidades corretas e o entrelaçamento quântico caso exista.

A figura seguinte mostra os resultados das medições feitas, para o mesmo exemplo anterior:

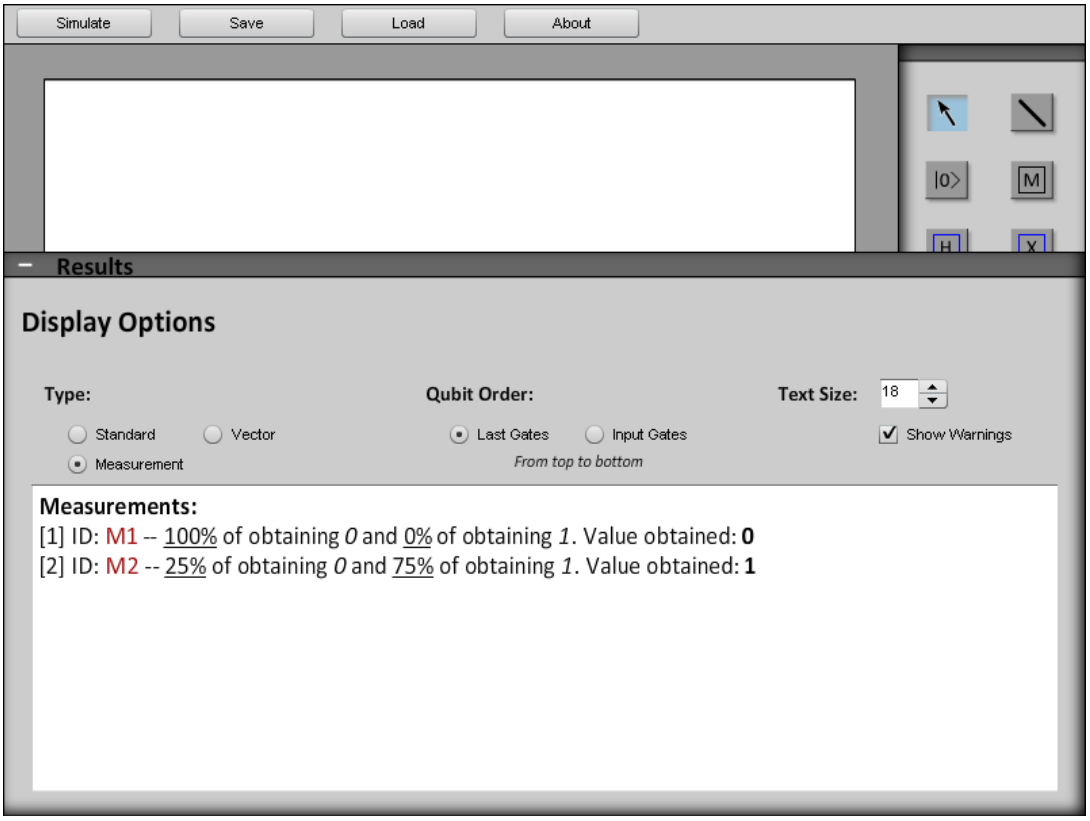

Figura 4.5 - Exemplo da janela de medição

4.6 – Funcionalidades de *Save* e *Load*

Foram criadas funcionalidades para gravar e abrir os circuitos tal como foram concebidos pelo utilizador, bem como qualquer matriz adicionada à ferramenta de portas personalizadas.

Depois de o utilizador carregar na opção de *Save* irá aparecer uma nova janela como a seguinte:

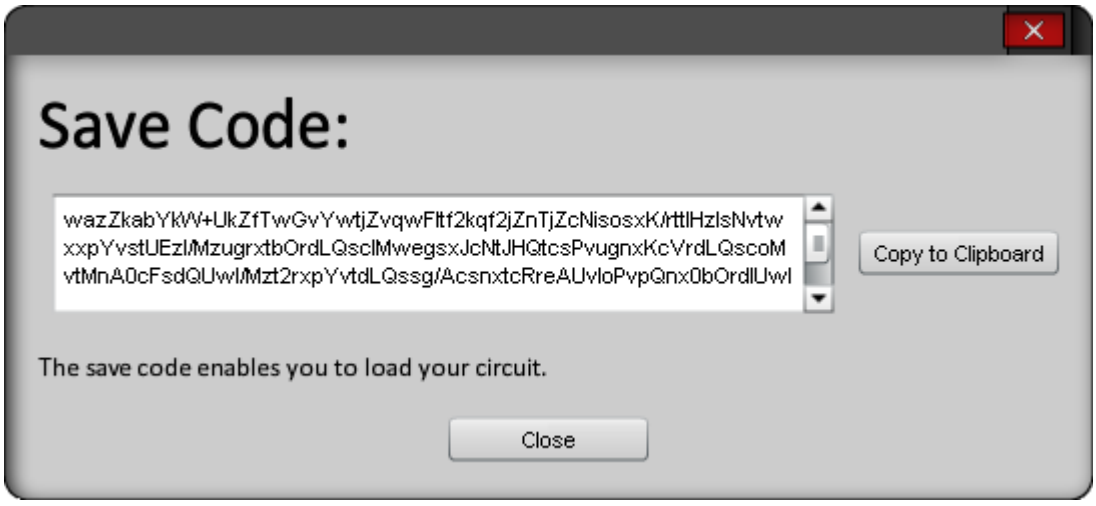

Figura 4.6 - Janela de *Save* do protótipo

O programa irá criar um código que traduz as especificações do circuito a gravar, a este será acrescentado uma verificação especial encriptada, com o intuito de o programa poder confirmar se o código não contem erros posteriormente, quando o utilizador pretender abrir novamente o circuito através da opção *Load*.

Se o utilizador carregar na opção "*Copy to Clipboard*" o código será automaticamente gravado para a área de transferência, tal como se tivesse sido utilizado o atalho do teclado CTRL + C.

O utilizador pode então guardar este código num documento de texto ou qualquer outro ficheiro para o poder utilizar posteriormente.

Da mesma forma, caso o botão de *Load* seja premido, surgirá uma nova janela, semelhante à anterior, onde o utilizador poderá introduzir o código anterior para restaurar novamente o seu circuito.

4.7 – Testes de verificação

Para averiguar que o programa está a funcionar corretamente e a obter os resultados corretos, foram efetuados vários testes durante o seu desenvolvimento. Nesta secção serão apresentados alguns dos testes e respetivos resultados obtidos.

4.7.1 – Exemplo de preparação de um dos estados Bell

A figura seguinte representa a preparação de um dos estados de Bell:

$$\left|\psi^{+}\right\rangle = \frac{\left|01\right\rangle + \left|10\right\rangle}{\sqrt{2}}. \tag{4.2}$$

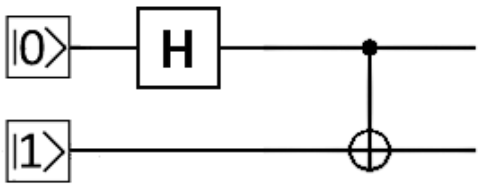

Figura 4.7 – Exemplo da preparação de um estado de Bell

O resultado obtido na notação Dirac está expressa na figura seguinte:

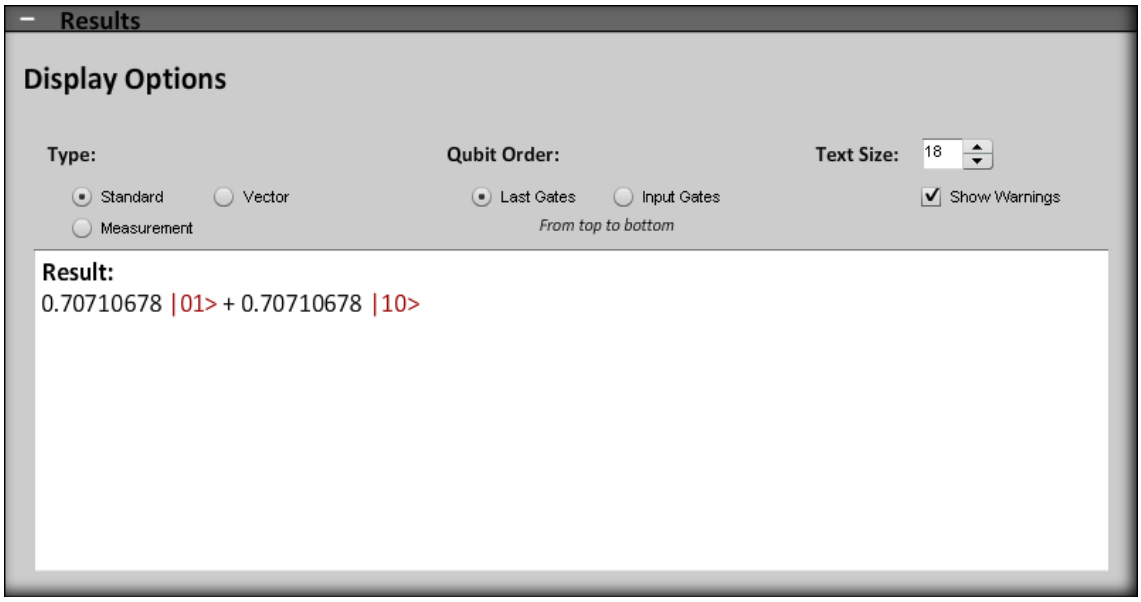

Figura 4.8 – Resultado do exemplo de preparação de um dos estados de Bell

4.7.2 – Exemplo da transformada de Fourier

O circuito da figura seguinte aplica a transformada de Fourier ao estado inicial $\psi = |010\rangle$.

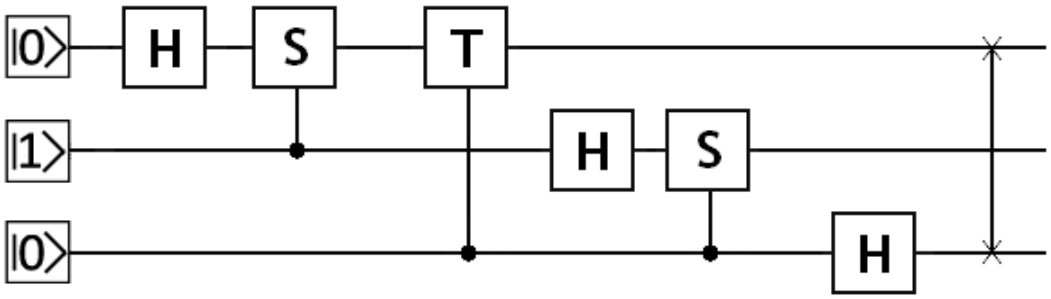

Figura 4.9 – Exemplo da transformada de Fourier

As portas S e T correspondem a mudanças de fase e as suas matrizes são dadas respectivamente pelas equações (4.3) e (4.4)

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{4.3}$$

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \tag{4.4}$$

O resultado obtido na forma matricial está visível na seguinte figura

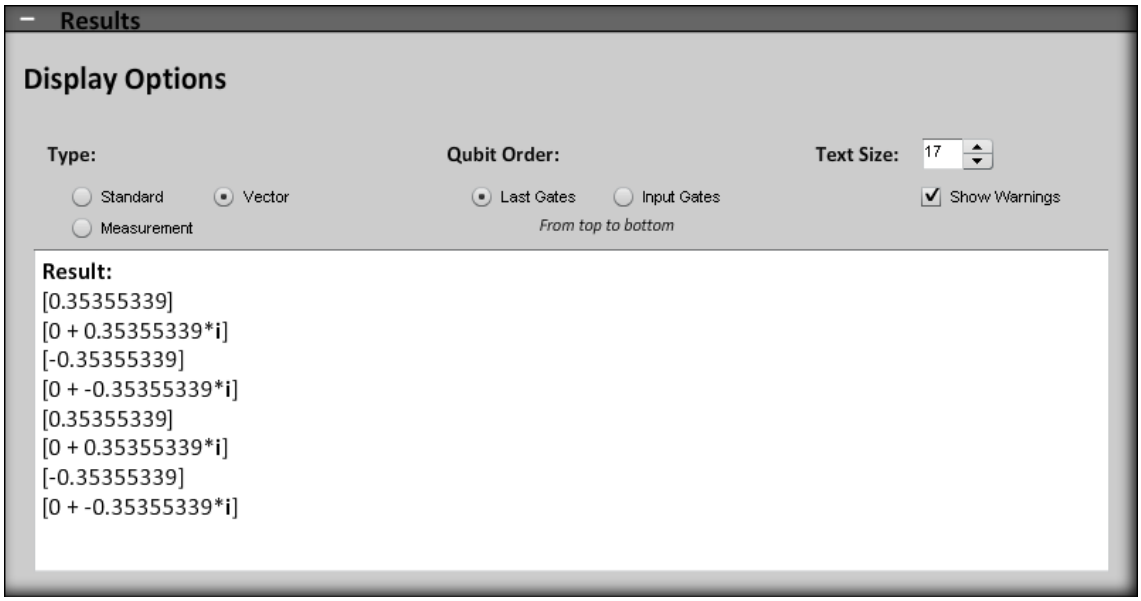

Figura 4.10 – Resultado obtido para a transformada de Fourier

4.7.3 – Exemplo do algoritmo de Grover

O circuito da figura seguinte representa o algoritmo de Grover para $N = 4$ e $f(x) = 1$ para $x = 3$.

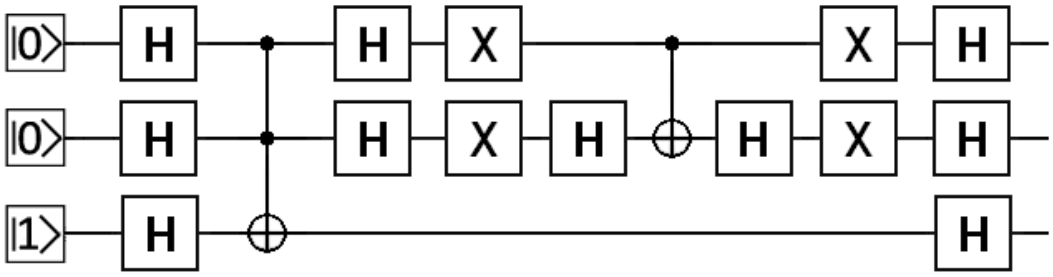

Figura 4.11 – Algoritmo de Grover para N=4

O circuito da figura seguinte representa o algoritmo de Grover para $N = 4$.

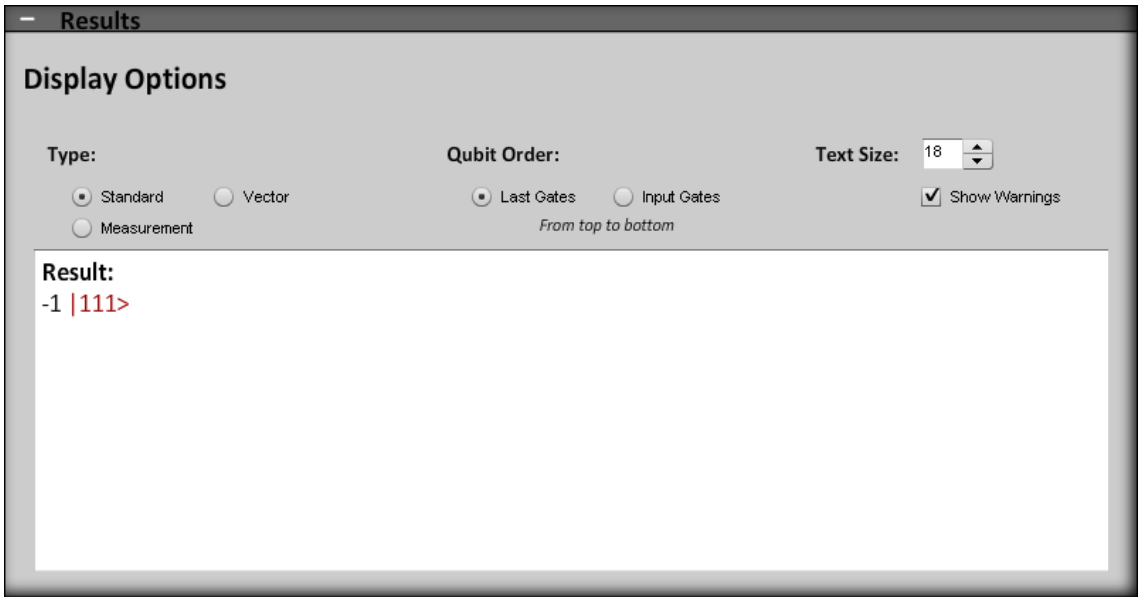

Figura 4.12 – Resultado obtido para o algoritmo de Grover com N=4

Capítulo 5

Conclusões e implementações futuras

Este capítulo sumaria o desenvolvimento da dissertação e as conclusões obtidas, assim como sugere algumas implementações que podem ser feitas futuramente de modo a melhorar o trabalho realizado.

5.1 – Desenvolvimento

Os objetivos propostos para este trabalho foram cumpridos. Começou por ser feita uma revisão bibliográfica da literatura existente e um estudo aprofundado das áreas envolvidas no tema, tais como a teoria da informação, teoria da computação e a mecânica quântica.

No entanto, durante este processo surgiu a necessidade de utilizar um programa de simulação de circuitos quânticos, porém os programas disponíveis não satisfaziam os requisitos procurados. Como tal surgiu a oportunidade de criar um programa de simulação de circuitos quânticos, uma vez que este é útil para a conceção de algoritmos quânticos.

Desta forma, o objetivo deste trabalho passou primariamente a ser o desenvolvimento de um programa que permita ao utilizador criar, editar e simular circuitos quânticos.

Foi então desenvolvida uma interface gráfica para a aplicação, com o intuito de ser fácil de usar e visualizar os circuitos, bem como estes poderem ser rapidamente criados ou editados.

O programa possui várias das portas mais importantes predefinidas, mas também foi adicionada a funcionalidade de o utilizador poder criar as suas portas personalizadas.

Finalmente foi adicionada a possibilidade de se gravar os circuitos criados, para poder posteriormente restaura-los, continuando assim a simulação.

A aplicação pode ser descarregada para o computador ou acedida através de uma página *web*, permitindo que qualquer pessoa nele interessada o possa usar.

Assim, o protótipo perfaz todos os requisitos inicialmente estruturados, podendo ser uma ferramenta de auxílio a investigadores da área, tal como para fins de aprendizagem e ensino.

5.2 – Conclusões

Uma vez que um computador clássico consegue simular um computador quântico, hipoteticamente não existiria nenhum problema que não fosse possível resolver num computador clássico, admitindo recursos infinitos, tais como memória ou tempo para completar o algoritmo, ou seja uma máquina de Turing [21]. No entanto, na prática não podemos contar com recursos infinitos, como tal, existem problemas que não são possíveis de completar devido à limitação da memória do computador ou porque demorariam demasiado tempo, como o exemplo da factorização de números grandes, que pode levar milhões de anos a terminar.

Através da computação quântica, muitos destes problemas tornam-se exequíveis, uma vez que os resolveriam eficientemente, com o algoritmo correto. Mas criar um algoritmo quântico que seja mais eficiente que o algoritmo clássico correspondente não é fácil. Apesar de ser possível tirar partido do paralelismo quântico através da sobreposição, para computar 2^n estados em simultâneo, não é possível extrair mais do que n bits de informação devido ao problema da medição colapsar o estado do sistema. Desta forma, é necessário criar um algoritmo que execute a computação tirando partido do paralelismo quântico e, no final, tenha preparado o sistema de tal modo que seja possível extrair a informação procurada com a medição.

Apesar das dificuldades, já existem alguns algoritmos que demonstram o aumento da eficiência da computação quântica quando comparada à clássica. A factorização de números através do algoritmo de Shor, a procura de elementos numa lista não ordenada com o algoritmo de Grover, ou a simulação de sistemas quânticos são todos exemplos de problemas onde a computação quântica é significativamente mais eficiente. Outro tipo de problemas em que a computação quântica deverá ser mais eficiente são os problemas de otimização. Estes englobam, entre outros, a escolha da melhor rota a usar, reconhecimento de imagens, análise da sequência do genoma.

Para além da computação quântica melhorar a eficácia de alguns algoritmos, também ajudou a desenvolver outras áreas como a criptografia quântica, de modo a poder transmitir informação de forma ainda mais segura.

Desta forma, pode-se verificar que a computação quântica é uma área muito promissora, com várias aplicações para o futuro.

5.3 – Implementações futuras

Embora o programa tenha cumprido os objetivos propostos, é sempre possível melhorar e implementar mais funções no futuro que sejam úteis:

- A eficiência das operações matemáticas entre matrizes, como a multiplicação, pode ser significativamente melhorada, reduzindo assim o tempo de execução para sistemas com um elevado número de qubits.

- Podem ser feitas melhorias ao nível da interface, de modo a facilitar a utilização pela parte do utilizador.

- Implementação de uma nova ferramenta que simule condições (se obter o valor x na medição y executar a porta z), deste modo passa a ser possível simular o protocolo de teleportação quântica.

- Na construção prática de computadores quânticos e dos seus componentes, existem sempre erros, uma vez que os sistemas não são ideais, como tal pode ser implementado um sistema de adição de erro à simulação.

Referências:

[1] G. E. Moore, (1998). “Cramming more components onto integrated circuits”. *Proceedings of the IEEE* 86, 82–85.

[2] J. R. Powell, (2008). “The Quantum Limit to Moore's Law”. *Proceedings of the IEEE* 96, 1247–1248.

[3] R. K. Cavin, P. Lugli e V. V. Zhirnov, (2012). “Science and Engineering Beyond Moore's Law”. *Proceedings of the IEEE* 100, 1720–1749.

[4] M. A. Nielsen e I. L. Chuang, (2000). “Quantum Computation and Quantum Information”. Cambridge University Press.

[5] N. David Mermin, (2007). “Quantum Computer Science: An Introduction”. Cambridge University Press.

[6] M. Hirvensalo, (2004). “Quantum Computing (Natural Computing Series)”. Springer.

[7] P. Kaye, R. Laflamme e M. Mosca, (2007). “An Introduction to Quantum Computing”. Oxford University Press.

[8] C. Bennett e S. J. Wiesner, (1992). “Communication via one- and two-particle operators on Einstein-Podolsky-Rosen states”. *Physical Review Letters* 69, 2881–2884.

[9] C. H. Bennett, et al., (1993) “Teleporting an Unknown Quantum State via Dual Classical and Einstein-Podolsky-Rosen Channels”. *Physical Review Letters* 70, 1895–1899.

[10] X. S. Ma, et al., (2012). “Quantum teleportation over 143 kilometres using active feed-forward”. *Nature* 489, 269–273.

[11] D. Deutsch e R. Jozsa, (1992). “Rapid solutions of problems by quantum computation”. *Proceedings of the Royal Society* 439, 553–558.

[12] P. W. Shor, (1994). “Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer”. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, 20–22.

[13] L. K. Grover, (1996). “A fast quantum mechanical algorithm for database search”. *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*, 212–219.

[14] R. Rivest, A. Shamir e L. Adleman, (1978). “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. *Communications of the ACM* 21, 120–126.

[15] C. H. Bennett e G. Brassard, (1984). “Quantum Cryptography: Public key distribution and coin tossing”. *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, 175–179.

[16] B. Butscher e H. Weimer, (2013). “libquantum. The C library for quantum computing and quantum simulation”. Disponível em: http://www.libquantum.de (acedido em fevereiro 2013).

[17] Kyriakos Sgarbas, Petroula Mavridi e Charalampos Tsimpouris, (2010). “Demo: Quantum Computer Simulator”. Disponível em: http://www.wcl.ece.upatras.gr/ai/resources/demo-quantum-simulation (acedido em fevereiro 2013).

[18] H. Watanabe, M. Suzuki e J. Yamazaki, (2011). “QCAD: GUI environment for quantum computer Simulator”. Disponível em: http://qcad.sourceforge.jp (acedido em fevereiro 2013).

[19] A. de Vries, (2010). “jQuantum - Quantum Computer Simulation Applet”. Disponível em: http://jquantum.sourceforge.net/jQuantumApplet.html (acedido em fevereiro 2013).

[20] Stat Owl, (2013). “Flash player version support”. Disponível em: http://www.statowl.com/flash.php (acedido em fevereiro 2013).

[21] A. M. Turing, (1937). “On Computable Numbers, with an Application to the Entscheidungsproblem”. *Proceedings of the London Mathematical Society* 42, 230–265.