Uma introdução à Computação Quântica

Wagner Jorcuvich Nunes da Silva

Trabalho de Formatura apresentado ao
Instituto de Matemática e Estatística da
Universidade de São Paulo

Curso: Bacharelado em Matemática Aplicada e Computacional

Orientador: Prof. Dr. Pedro da Silva Peixoto

São Paulo, novembro de 2018

Uma introdução à Computação Quântica

Esta versão da Monografia contém as correções e alterações sugeridas pela Banca Examinadora durante a apresentação da versão original do trabalho.

Banca Examinadora:

- Prof. Dr. Pedro da Silva Peixoto (orientador) - IME-USP
- Prof. Dr. Saulo Rabello Maciel de Barros - IME-USP
- Prof. Dr. Marcelo Keese Albertini - FACOM-UFU

Agradecimentos

Agradeço ao meu amigo e orientador, Prof. Pedro, por me dar essa oportunidade.
À Amerlende, Marcos e Adriana por terem paciência.

Resumo

JORCUVICH, W. N. S. **Uma introdução à Computação Quântica**. 2018.

A Computação Quântica é uma área relativamente recente e pouco conhecida, inclusive entre os cientistas da Computação. Surgiu na tentativa de físicos simularem sistemas regidos pela Mecânica Quântica por computadores clássicos, o que se mostrou inviável. Surge aí a necessidade de um novo modelo computacional que utiliza a estrutura quântica da matéria.

Este trabalho tem como objetivo principal estudar e compreender as noções básicas Computação Quântica. Para atingir tal objetivo, primeiramente são expostos os conhecimentos básicos da Mecânica Quântica através de uma linguagem simples, sem que haja um conhecimento prévio da área, de forma a remover a barreira inicial sobre o tema. Em seguida são apresentadas as estruturas que manipulam a entidade básica da computação quântica, o q-bit. São mostrados, também, alguns algoritmos, e simuladores.

Palavras-chave: mecânica quântica, computação quântica, q-bit, portas quânticas, simuladores quânticos.

Abstract

JORCUVICH, W. N. S. **An Introduction to Quantum Computing**. 2018.

Quantum Computation is a relatively recent and little known area, even among computer scientists. It arose in the attempt of physicists to simulate systems governed by quantum mechanics using classic computers, which proved impracticable. Thus, a need for a new computational model that uses the quantum structure of matter emerged.

This work has as main objective to study and to understand the basic concepts of Quantum Computing. In order to reach this goal, an introduction to Quantum Mechanics is presented in a simple language, without requiring prior knowledge of the area, in order to remove the initial barrier on the subject. Then, the structures to manipulate the basic entity of quantum computation, q-bit, are discussed. Some algorithms and simulators are also shown.

Keywords: quantum mechanics, quantum computation, q-bit, quantum gates, quantum simulators.

Sumário

Lista de Figuras **ix**

Lista de Tabelas **xi**

1 Introdução **1**

1.1 Desafios de Hilbert . . . 1
1.2 Máquina de Turing . . . 2
1.2.1 Descrição de uma máquina de Turing . . . 3
1.3 Lei de Moore . . . 4
1.4 Início da computação quântica . . . 4
1.5 Máquina de Turing quântica . . . 5
1.6 Algoritmos quânticos . . . 6
1.7 Desenvolvimento do hardware . . . 6

2 Noções de mecânica quântica **7**

2.1 Álgebra linear . . . 7
2.1.1 Espaços vetoriais . . . 8
2.1.2 Base e dimensão . . . 9
2.1.3 Subespaços vetoriais . . . 9
2.1.4 Produto interno . . . 9
2.1.5 Operadores lineares . . . 10
2.1.6 Produtos tensoriais . . . 11
2.2 Postulados da mecânica quântica . . . 12
2.3 Superposição . . . 14
2.4 Emaranhamento . . . 15
2.5 Teorema da não-clonagem . . . 15

3 Computação quântica **17**

3.1 Bit quântico (q-bit) . . . 17
3.1.1 Esfera de bloch . . . 18
3.2 Portas quânticas . . . 20
3.2.1 Portas quânticas de 1 q-bit . . . 21

3.2.2 Portas quânticas de múltiplos q-bits 26

4 Algoritmos e circuitos quânticos 31
4.1 Circuitos quânticos 31
4.2 Paralelismo quântico 32
4.2.1 Algoritmo de Deutsch 32
4.3 Soma aritmética Vedral 33
4.3.1 Exemplo de soma Vedral no circuito 34
4.4 Transformada de Fourier quântica 35
4.5 Soma aritmética Draper 36
4.5.1 Exemplo de soma Draper 37
4.6 Outros algoritmos quânticos 40

5 Simuladores de circuitos quânticos 43
5.1 Alguns simuladores 43
5.1.1 Simuladores *offline* 44
5.1.2 Simuladores *online* 48
5.2 Linguagens de programação 51
5.2.1 Imperativas 51
5.2.2 Funcionais 52

6 Conclusões 53
6.1 Considerações finais 53
6.2 Sugestões para pesquisas futuras 53

A Números complexos 55
A.1 Soma e multiplicação 55
A.2 Representação geométrica 56
A.3 Exponencial complexa 56
A.4 Argumento de um complexo 57

Referências Bibliográficas 59

Lista de Figuras

1.1 Esquema de uma Máquina de Turing. 3
1.2 Gráfico original da evolução dos transistores [Moo65]. 4
1.3 Esquema de uma Máquina de Turing Quântica [Meg08]. 5

3.1 Representação de um q-bit Esfera de Bloch [OS04]. 19
3.2 Representação, em circuitos, da Porta Pauli-I. 21
3.3 O estado do q-bit antes e depois da porta Pauli-X [Gra18]. 22
3.4 Representações, em circuitos, da Porta Pauli-X. 22
3.5 Representação, em circuitos, da Porta Pauli-Y. 23
3.6 Representações, em circuitos, da Porta Pauli-Z. 23
3.7 Representação, em circuitos, da Porta Hadamard. 24
3.8 Representação, em circuitos, da Porta de Fase 24
3.9 Representação, em circuitos, da Porta $\pi/8$ ou T. 25
3.10 Representação circuital da Porta CNOT Quântica. A linha superior representa o q-bit de controle, e a linha de baixo o q-bit-alvo 26
3.11 Representação circuital da Porta Toffoli Quântica. As linhas superiores representam os q-bits de controle, e a linha de baixo o q-bit-alvo. 27
3.12 Representação circuital da Porta Swap Quântica. 28
3.13 Representação circuital da Porta Fredkin Quântica. A linha superiore representam o q-bit de controle, e as linha de baixo os q-bits-alvo. 28
3.14 Representação circuital de uma Porta Oráculo. 29

4.1 Representação de um Circuito. 31
4.2 Representação de um Circuito de Deutsch. 32
4.3 Representação dos circuitos de Carry e Sum [VBE95]. 33
4.4 Representação do circuito de Soma de Vedral [VBE95], adaptado por Kowada [Kow06]. 34
4.5 Representação de um Circuito de soma Vedral para doi bits. 35
4.6 Representação do circuito de Transformada de Fourier Quântica [NC00]. . . 36
4.7 Representação do circuito de Soma Draper [Dra00]. 37
4.8 Representação do circuito de Soma Draper para dois bits (1+2). 40

5.1 Imagem da interface do aplicativo QCS, com um circuito de SWAP [Pla]. . . 44
5.2 Imagem da interface do aplicativo QCS, com sa solução do circuito SWAP [Pla]. 45
5.3 Imagem da interface do Simulador Zeno, com um circuito somador Vedral para 3 q-bits, 1+2. 45
5.4 Tela do resultado, tipo ket, da soma Vedral. 46
5.5 Imagem da interface do Simulador Zeno, com um circuito de TFQ para 3 q-bits. 46
5.6 Tela do resultado, tipo ket, do Zeno para TFQ. 47
5.7 Tela do resultado, tipo histograma, do Zeno para TFQ [Cab04]. 47
5.8 Imagem da interface do IBM-Q, com um circuito de TFQ para 2 q-bits [Qa]. 48
5.9 Imagem da interface do IBM-Q, com o resultado da TFQ para 2 q-bits [Qa]. 49
5.10 Imagem da interface do Quirk, TFQ para 3 a-bits, com entrada $|000\rangle$. 50
5.11 Imagem da interface do Quirk com soma vedral para 2 q-bits, 1+2. 50

Lista de Tabelas

2.1 A notação de *Dirac*, utilizada em mecânica quântica para conceitos de álgebra linear [NC00]. 8

3.1 Direção e sentido do vetor representativo do q-bit para alguns estados.[OS04] 20

Capítulo 1

Introdução

Nos últimos 50 anos, os computadores deixaram de ser do tamanho de salas e deixaram de ser objetos apenas de empresas e privilegiados para estarem à mão de todos, cada vez mais potentes e compactos. Em parte, esta transformação foi possível graças à miniaturização dramática dos componentes básicos de um computador. Esta tendência foi identificada e junto com ela as limitações de ordem física.

Quase concomitante ao desenvolvimento da eletrônica, desencadeadora da construção de computadores, desenvolveu-se, também, a física moderna, onde se afere que sistemas físicos como átomos e partículas de menores grandezas, comportam-se de maneiras muito diferente de objetos do dia a dia. Na verdade, eles são governados pelas leis da *mecânica quântica* em vez de mecânica clássica.

No início dos anos 80, alguns estudiosos da área começaram a questionar o que significaria um computador operar na escala de um átomo por bit. As operações elementares de tal computador precisariam ser descritas em termos da mecânica quântica.

Físicos e cientistas da computação entenderam que certos efeitos quânticos permitem tipos inteiramente novos de tarefas a serem executadas. Nasce aí um novo paradigma na tecnologia, uma nova área que envolve a matemática, a ciência da computação, a física moderna e engenharias correlatas.

A respeito da rapidez no desenvolvimento do computador clássico, o computador quântico poderá não demorar muito a ser realidade tangível a todos, por isso é necessária a compreensão de que divulgação dessa nova área é o desafio atual.

1.1 Desafios de Hilbert

Em 1900, em uma palestra intitulada *"Mathematische Probleme"*, antes do segundo congresso internacional de matemáticos realizado em Paris, David Hilbert (1862 – 1943), profundamente envolvido com os fundamentos da matemática e com a possibilidade teórica de soluções por meios mecânicos, postulou 23 problemas de temas diversos em matemática e áreas afins para serem resolvidos no século XX. O "décimo problema"é sobre equações

Diofantinas, conhecido como o problema de decisão, ou "Entscheidungsproblem".

"Dada uma equação Diofantina, com coeficientes inteiros com um número qualquer de variáveis, é possível elaborar um processo que decida, através de um número finito de operações, se a equação tem soluções inteiras." [Hil02].

Hoje a expressão "conceber um processo" tem um sentido de encontrar um algoritmo, mas quando os problemas foram propostos, não havia nenhuma noção matematicamente rigorosa para o conceito de algoritmo. Em uma linguagem mais atual: existe um algoritmo que, dada uma equação diofantina, determina se esta tem ou não solução?

Na proposta do décimo problema de Hilbert o foco de interesse aqui é apenas na existência das soluções e não na obtenção explícita destas soluções.

Na conferência internacional de matemática de 1928, juntamente com seu orientado Wilhelm Ackermann (1896 – 1962), estabeleceu outras três influentes questões que tinham relação direta com o décimo problema [HA28]:

- Seria a matemática completa, no sentido de que toda e qualquer afirmativa matemática possa ser sempre provada ou negada?
- Seria a matemática consistente, no sentido de que jamais se possa chegar a uma inconsistência por meio de raciocínio logicamente correto?
- Poderia a matemática decidir por si só, no sentido de que existe um procedimento mecânico capaz de determinar a falsidade ou a veracidade de qualquer afirmação?

Kurt Friedrich Gödel, já em 1931, trouxe contribuições respondendo negativamente às duas primeiras, ao provar que não é possível utilizar uma teoria matemática para demonstrar sua própria consistência e que dada uma teoria matemática, sempre existirão questões cujas as provas ou negações estarão fora do alcance daquela teoria. São teoremas que, de certa forma, estabelecem os limites da própria matemática, chamados de teoremas da incompletude de Gödel [Gö31]. Com palavras muito simples, pode-se dizer que estes teoremas afirmam que sempre existirão fatos verdadeiros que não podem ser matematicamente demonstrados.

Os problemas postulado por Hilbert precede invenção de computadores em décadas. Foi apenas nos anos 30 que tais questões foram formuladas e tratadas dentro do que ficou depois conhecido como teoria da computabilidade.

1.2 Máquina de Turing

Em 1936 Alan Mathison Turing (1912 - 1954) deu início à fundamentação das bases teóricas da ciência da computação, mostrando que era possível executar operações computacionais sobre a teoria dos números por meio de uma máquina que tivesse embutidas as regras de um sistema formal. Embora propriamente não existisse tal máquina, ele enfatizou desde o início que tais mecanismos poderiam ser construídos. Seu trabalho abriu uma nova perspectiva no esforço de formalizar a matemática e a computação.

Esta máquina hipotética ficou conhecida mais tarde como *Máquina de Turing*. Foi em um artigo intitulado *"On Computable Numbers, with an application on the Entscheidungsproblem"* que esta teoria foi estabelecida pela primeira vez, dando uma resposta ao 10º problema de Hilbert [Tur36]. Ao unir matemática e lógica na forma de uma máquina, Turing tornou possíveis sistemas processadores de símbolos.

Hoje todos os computadores são construídos tendo como base o modelo da Máquina de Turing.

1.2.1 Descrição de uma máquina de Turing

O processo computacional foi graficamente mostrado por Turing que considerou um dispositivo que pudesse ler e escrever símbolos em uma fita que estava dividida em quadrados. Uma cabeça de leitura e gravação se moveria em qualquer direção ao longo da fita, um quadrado por vez, e uma unidade de controle poderia interpretar uma lista de instruções simples sobre leitura e gravação de símbolos nos quadrados, movendo-se, ou não, para a direita ou esquerda. O quadrado que é lido em cada etapa é conhecido como "quadrado ativo". A regra que está sendo executada determina o que se convencionou chamar *estado* da máquina. A fita é potencialmente infinita [Sip05].

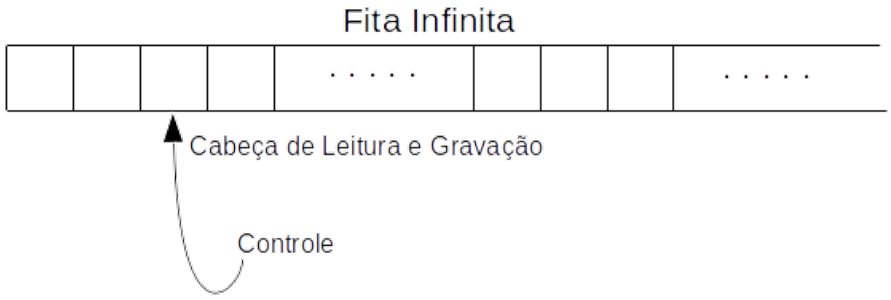

Figura 1.1: *Esquema de uma Máquina de Turing.*

Cada instrução estabelece uma ação a ser executada. No caso, são estabelecidos quatro diferentes tipos de regra:

- Substituir branco por símbolo;
- Substituir símbolo por branco;
- Mover um quadrado para a direita;
- Mover um quadrado para a esquerda.

A máquina recebe um conjunto de instruções e executa. O dispositivo pode mover a cabeça de leitura e gravação para a direita ou esquerda, ler a posição, substituir um o branco por símbolo, símbolo por branco ou apenas mover a cabeça novamente.

1.3 Lei de Moore

No ano de 1965, Gordon Earl Moore (1929 -), tornou-se cofundador e presidente da Intel e constatou que a complexidade para construção de componentes (transistores), a um custo mínimo, tem aumentado aproximadamente um fator a cada dois anos. Em outras palavras, o número de transistores dos chips tem um aumento de 100%, pelo mesmo custo, a cada período de 2 anos [Moo65]. Essa previsão ficou conhecida com a Lei de Moore que mais tarde foi revista para um período que seria a cada 18 meses.

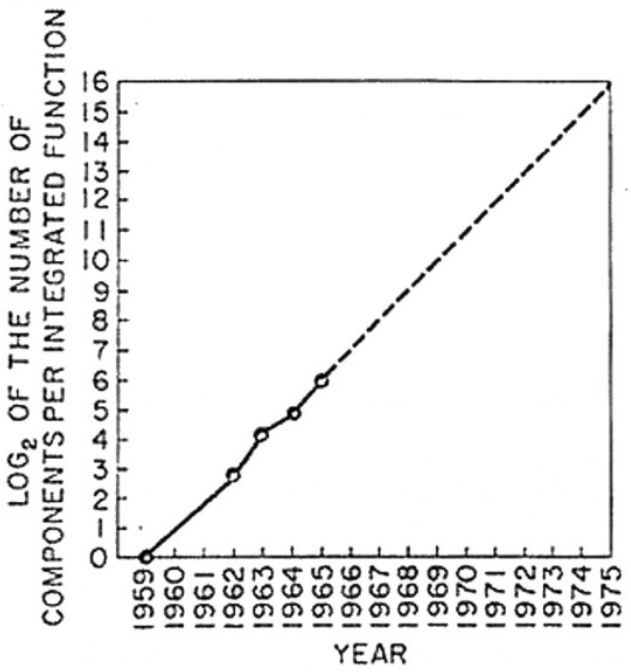

Figura 1.2: *Gráfico original da evolução dos transistores [Moo65].*

Uma das consequências da lei de Moore é a corrida das indústrias de semicondutores, que investiam significativamente seus recursos para seguir a previsão. Sem ela talvez não existisse um nível tão acelerado do desenvolvimento dos hardwares em um curto período.

1.4 Início da computação quântica

Pode-se considerar o marco inicial da computação quântica o ano de 1981 com o físico Richard Philips Feynman (1918 – 1988) elaborando a primeira proposta de um dispositivo que se utiliza de fenômenos quânticos para executar rotinas computacionais.

Feynman argumentou que computadores clássicos podem simular a física clássica, porém o mesmo não ocorre com a física quântica, uma vez que a dimensão do espaço nela tratado, cresce exponencialmente em função do número de partículas acrescentadas ao sistema. Ele,

então, questionou se um dispositivo que usasse as leis da mecânica quântica, para realizar cálculos, não poderia simular eficientemente a mecânica quântica [Fay82].

Vale observar que é relativamente fácil modelar o comportamento de um átomo em um computador clássico, porém um gás em uma sala possui algo em torno de 10 elevado a 26 átomos, o que precisaria de algo em torno de bilhões e bilhões de gigabytes de memória para simular corretamente por vias clássicas [OS04].

1.5 Máquina de Turing quântica

Em 1985 David Deutsch (1953 -) cria o primeiro algoritmo quântico e faz uma descrição do equivalente quântico para a máquina de turing [Deu85].

As funções realizadas em uma máquina de Turing Quântica ocorrem via interações quânticas. A fita e a cabeça em si existem em um estado quântico. No lugar da célula, a máquina de Turing Quântica abriga os q-bits, que apresentam estados de superposição de 0 ou 1. Ela pode codificar muitas entradas para um problema simultaneamente e calcular todas as entradas ao mesmo tempo.

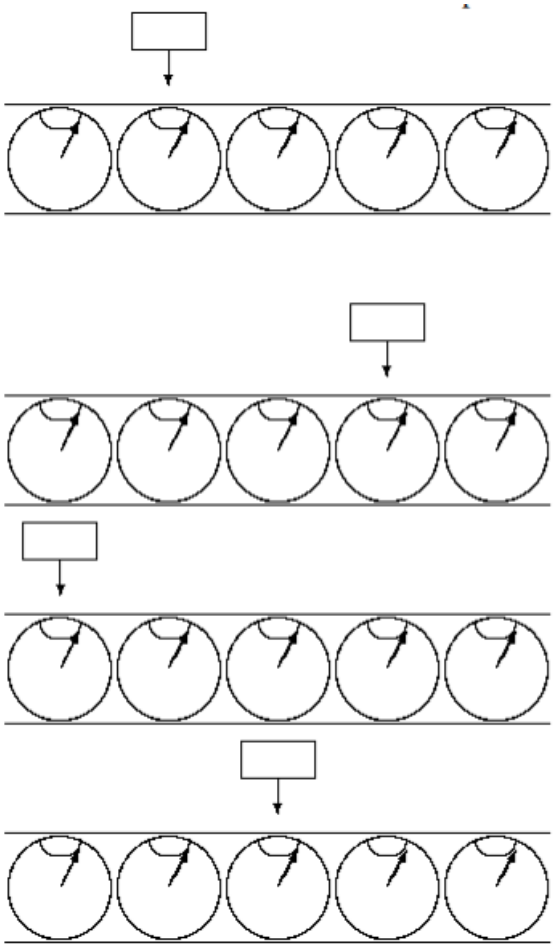

Figura 1.3: *Esquema de uma Máquina de Turing Quântica [Meg08].*

Observa-se na figura 1.3 o primeiro esquema representando o estado inicial da fita e da

cabeça. Os desenhos seguintes representam as posições diferentes que a cabeça se movimentam, podendo apresentar um estado superposto desses três estados ao mesmo tempo.

1.6 Algoritmos quânticos

Em 1994 Peter Shor mostra que problemas considerados intratáveis por computadores clássicos, como a fatoração de números primos, podem ser resolvidos com computadores quânticos.

Grande parte da ciência da computação é fundamentada na teoria dos números e as propriedades dos números primos. A segurança digital se baseia em códigos de criptografia, embasados na fatoração de números primos.

Shor revoluciona o mundo da computação com o seu algoritmo, diminuindo o tempo de fatoração [Sho94]. Grover em 1996 demonstra, que em computação quântica, o tempo para se encontrar um elemento é substancialmente menor em relação à clássica. [Gro96]

1.7 Desenvolvimento do hardware

Os primeiros protótipos de computador quântico apareceram em 1999 , no *MIT (Massachusettes Institute of Technology)*. Em 2007, a empresa canadense $D-Wave$ anunciou a construção do primeiro processador quântico do mundo, o Orion, com capacidade de processamento de 16 q-bits. Na sequência IBM e Google se adiantaram também no desenvolvimento.

Em novembro de 2017, a IBM anunciou sucesso no desenvolvimento de um computador quântico de 50 q-bits. E em janeiro de 2018, foi a vez da Intel (49 q-bits), e em março a Google (72 q-bits).

A capacidade computacional de um computador quântico pode ser impressionante, por exemplo, um processador de 100 q-bits seria mais poderoso do que a soma de todos os computadores atuais no planeta.

Nenhuma empresa propôs formalmente a entrada de seus computadores quânticos no mercado. A perspectiva é que os computadores quânticos entrem no mercado não para substituir completamente os PCs tradicionais, mas para integrá-los em um sistema mais poderoso.

Sua real eficácia foi comprovada recentemente pela IBM, mostrando que computadores quânticos são muito mais rápidos do que os modelos tradicionais na resolução de alguns problemas [BGK18].

Capítulo 2

Noções de mecânica quântica

Por meio de experimentos no início do século XX, cientistas observaram que as leis clássicas não eram aplicáveis a objetos muito pequenos. Em outras palavras, o que havia sido calculado pela mecânica clássica não refletia o comportamento de objetos extremamente pequenos. A partir de então uma nova teoria que descrevendo o comportamento de objetos microscópicos teve de ser construída, a Mecânica Quântica. [CG17]

A mecânica quântica é a teoria que descreve o comportamento físico de sistemas microscópicos, como fótons, átomos e moléculas. Qualquer sistema com tamanho na escala de Angstrons ($1\text{Å} = 10^{-10}m$) sofre a influência de efeitos quânticos.

Matematicamente falando, a mecânica quântica é uma teoria, pois é regida por um conjunto de axiomas (princípios). As consequências desses axiomas descrevem o comportamento dos sistemas da mecânica quântica.

Para descrever estas situações pouco usuais com precisão, são necessárias ferramentas matemáticas diferentes daquelas usadas na mecânica clássica. Além de noções de números complexos e operação com matrizes, é necessária uma base de álgebra linear.

2.1 Álgebra linear

Seguem algumas noções básicas sobre espaços vetoriais e produtos internos que serão muito utilizadas no decorrer do texto. Principalmente espaços vetoriais complexos, que aparecem naturalmente na mecânica quântica. Será utilizada a notação-padrão de *Dirac*. Como segue a tabela 2.1.

Os fundamento de álgebra linear, aqui descritos, foram extraídos principalmente do livro da Amaral [ABC11].

Notação	Descrição
z^*	Conjugado complexo de z: $(1+i)^* = 1-i$
$\|\psi\rangle$	Vetor. Também chamado de *ket*
$\langle\psi\|$	Vetor dual de $\|\psi\rangle$. Também chamado de *bra*
$\langle\phi\|\psi\rangle$	Produto escalar entre $\|\phi\rangle$ e $\|\psi\rangle$
$\|\phi\rangle \otimes \|\psi\rangle$	Produto tensorial entre $\|\phi\rangle$ e $\|\psi\rangle$
$\|\phi\rangle\|\psi\rangle$	Notação abreviada de produto tensorial
A^*	Complexo conjugado da matriz A
A^T	Transposta da matriz A
$A^\dagger$	Conjugado hermitiano, ou matriz adjunta de A, $A^\dagger = \left(A^T\right)^*$
$\langle\phi\|A\|\psi\rangle$	Produto escalar entre $\|\phi\rangle$ e $A\|\psi\rangle$

Tabela 2.1: *A notação de Dirac, utilizada em mecânica quântica para conceitos de álgebra linear [NC00].*

2.1.1 Espaços vetoriais

Definição 1 *Um espaço vetorial V sobre um corpo C é um conjunto, cujos elementos são chamados de vetores e denotados por $|\cdot\rangle$, munido de uma soma vetorial:*

$$\begin{aligned} + : V \times V &\to V \\ (|\psi\rangle, |\phi\rangle) &\mapsto |\psi\rangle + |\phi\rangle \end{aligned} \tag{2.1}$$

e de um produto por escalar:

$$\begin{aligned} \cdot : C \times V &\to V \\ (\alpha, |\psi\rangle) &\mapsto \alpha|\psi\rangle \end{aligned} \tag{2.2}$$

Dado que para todos $|\psi\rangle, |\varphi\rangle, |\phi\rangle \in V$ e $\alpha, \mu \in C$:

1. (Associatividade) $|\psi\rangle + (|\varphi\rangle + |\phi\rangle) = (|\psi\rangle + |\varphi\rangle) + |\phi\rangle$ e $\alpha\left(\beta|\psi\rangle\right) = (\alpha\beta)\,|\psi\rangle$;
2. (Comutatividade) $|\psi\rangle + |\phi\rangle = |\phi\rangle + |\psi\rangle$;
3. (Existência de zero) Existe vetor $0 \in V$ tal que $|\psi\rangle + 0 = |\psi\rangle$;
4. (Existência do vetor oposto) Dado $|\psi\rangle \in V$ existe um vetor $-|\psi\rangle \in V$ tal que $|\psi\rangle + (-|\psi\rangle) = 0$;
5. (Distributividade) $\alpha\left(|\psi\rangle + |\phi\rangle\right) = \alpha|\psi\rangle + \alpha|\phi\rangle$ e $(\alpha+\beta)\,|\psi\rangle = \alpha|\psi\rangle + \beta|\psi\rangle$;
6. $1|\psi\rangle = |\psi\rangle$ quando 1 é a unidade da multiplicação no corpo C.

2.1.2 Base e dimensão

Definição 2 *Uma expressão do tipo*

$$\alpha_1|\phi_1\rangle + \cdots + \alpha_k|\phi_k\rangle, \alpha_i \in C \tag{2.3}$$

é uma combinação linear dos vetores $|\phi_1\rangle, \cdots, |\phi_k\rangle$*. Dado um conjunto de vetores, ele gera* V *se todo elemento de* V *pode ser escrito como combinação linear dos elementos desse conjunto.*

Definição 3 *Um conjunto de vetores* $\{|\phi_1\rangle, \cdots, |\phi_k\rangle\} \subset V$ *é linearmente independente (LI) se a equação*

$$\alpha_1|\phi_1\rangle + \cdots + \alpha_k|\phi_k\rangle = 0 \tag{2.4}$$

só admite a solução trivial $(\alpha_1, \cdots, \alpha_k) = (0, \cdots, 0)$*. Caso contrário, os vetores são linearmente dependentes (LD).*

Uma base para um espaço vetorial é definida em [NC00] como um conjunto de vetores linearmente independentes que geram o espaço. A grosso modo pode se entender a definição de base como um conjunto mínimo de vetores do espaço que pode criar todos os outros vetores deste espaço através de combinação linear.

2.1.3 Subespaços vetoriais

Um subespaço vetorial S do espaço vetorial V é um subconjunto de V que é, ele mesmo, um espaço vetorial com as operações de soma e multiplicação por escalar definidas em V. As seguintes propriedades devem ser satisfeitas:

- $0 \in S$**;**
- $|\psi\rangle + |\phi\rangle \in S$ para todo par $|\psi\rangle, |\phi\rangle \in S$**;**
- $\alpha|\psi\rangle \in S$ para todo $\alpha \in C$ e todo $|\psi\rangle \in S$**.**

2.1.4 Produto interno

Definição 4 *Um espaço vetorial sobre* $\mathbb{C}$*, o conjuntos dos números complexos, é chamado espaço vetorial complexo.*

Definição 5 *Dado* V *um espaço vetorial complexo, o* produto interno *é uma aplicação:*

$$\begin{aligned} \langle\cdot|\cdot\rangle : V \times V &\to \mathbb{C} \\ (|\psi\rangle, |\phi\rangle) &\mapsto \langle\psi|\phi\rangle \end{aligned} \tag{2.5}$$

satisfazendo as seguintes propriedades: para todo $|\psi\rangle, |\phi\rangle, |\varphi\rangle \in V$ *e* $\alpha, \beta \in \mathbb{C}$

1. $\langle\alpha\psi + \beta\varphi|\phi\rangle = \alpha\langle\psi|\phi\rangle + \beta\langle\varphi|\phi\rangle$;

2. $\langle\psi|\phi\rangle = \langle\phi|\psi\rangle^*$;

3. $\langle\psi|\psi\rangle \geq 0$;

4. Se $\langle\psi|\psi\rangle = 0 \Leftrightarrow |\psi\rangle = 0$.

Por exemplo, $\mathbb{C}^n$ tem produto interno definido por:

$$\langle (y_1, \cdots, y_n) \,|\, (z_1, \cdots, z_n)\rangle = \sum_i y_i^* z_i = [y_1^* \cdots y_n^*] \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \tag{2.6}$$

O produto interno permite introduzir uma noção que generaliza a um espaço vetorial qualquer a ideia de perpendicularidade no espaço.

Definição 6 *Dois vetores $|\psi\rangle$ e $|\phi\rangle$ são* ortogonais *se $\langle\psi|\phi\rangle = 0$. E, um conjunto $E = \{|\phi_1\rangle, \cdots, |\phi_k\rangle\}$ é ortogonal se seus elementos são dois a dois ortogonais, ele é ortonormal se é ortogonal e $\langle\phi_i|\phi_i\rangle = 1$ para todo i.*

O Produto Interno também denefine a norma de um vetor: $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$.

Definição 7 *Um espaço vetorial* completo $\mathcal{H}$ *com Produto Interno $\langle\cdot|\cdot\rangle$ e a norma $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$, é chamado* Espaço de Hilbert.

Um espaço vetorial é dito completo se todas as Sequências de Cauchy convergem dentro do espaço. Para aprofundar nesse assunto, a sugestão é o livro do Lima [Lim05].

2.1.5 Operadores lineares

Definição 8 *Sejam V e W espaços vetoriais sobre o corpo $\mathbb{C}$ e A uma aplicação $A : V \to W$. A é chamada de* operador linear *(ou transformação linear) em V, se dados $|\psi\rangle, |\phi\rangle \in V$ e $\alpha, \beta \in \mathbb{C}$:*

$$A(\alpha|\psi\rangle + \beta|\phi\rangle) = \alpha A(|\psi\rangle) + \beta A(|\phi\rangle) = \alpha A|\psi\rangle + \beta A|\phi\rangle \tag{2.7}$$

O operador linear é um endomorfismo quando $W = V$ ($A : V \to V$).

Definição 9 *Se $A : V \to V$ é uma transformação linear, então pode-se procurar vetores não nulos satisfazendo, para algum $\alpha \in C$, a equação*

$$A|\psi\rangle = \alpha|\psi\rangle \tag{2.8}$$

As soluções $|\psi\rangle$ são conhecidas como autovetores *e o respectivo α como* autovalor *de A.*

Definição 10 *Seja A um operador linear em um espaço de Hilbert $\mathcal{H}$, seu* operador conjugado hermitiano *é o operador $A^\dagger$ em $\mathcal{H}$, tal que quaisquer vetores $|\psi\rangle, |\phi\rangle \in V$ vale:*

$$(|\psi\rangle, A|\phi\rangle) = \left(A^\dagger|\psi\rangle, |\phi\rangle\right) \tag{2.9}$$

Um operador A cujo adjunto é o próprio A é chamado de *hermitiano* ou *auto-adjunto*. Ou seja, $A = A^\dagger$. Além disso é chamado de operador unitário se $A^\dagger A = AA^\dagger = I$, onde I é a matriz identidade.

Se um operador A é hermitiano, então $(|\psi\rangle, A|\phi\rangle) = (A|\psi\rangle, |\phi\rangle)$. Desse fato, pode-se usar a notação:

$$(|\psi\rangle, A|\phi\rangle) = \langle\psi|A|\phi\rangle \tag{2.10}$$

Da definição sai que $(AB)^\dagger = B^\dagger A^\dagger$, e por convenção $|\psi\rangle \equiv \langle\psi|$. Com isso fica fácil ver que $(A|\psi\rangle)^\dagger = \langle\psi|A^\dagger$ [NC00].

2.1.6 Produtos tensoriais

Para tratar do problema de várias partículas na mecânica quântica é necessário introduzir o conceito de produto tensorial em espaços de Hilbert.

Definição 11 *Sejam V e W espaços vetoriais de dimensões m e n respectivamente, com as seguintes bases $|\psi_1\rangle, \cdots, |\psi_m\rangle$ e $|\phi_1\rangle, \cdots, |\phi_n\rangle$. O espaço vetorial $V \otimes W$ é gerado por combinações lineares dos seguintes elementos $|\psi_i\rangle \otimes |\phi_j\rangle$, chamando $|\psi_i\rangle \otimes |\phi_i\rangle$ de base produto para $V \otimes W$.*

O produto tensorial satisfaz as seguintes propriedades:

1. Para um escalar α arbitrário, e elementos $|\psi\rangle$ de V e $|\phi\rangle$ de W:

$$\alpha\left(|\psi\rangle \otimes |\phi\rangle\right) = (\alpha|\psi\rangle) \otimes |\phi\rangle = |\psi\rangle \otimes (\alpha|\phi\rangle) \tag{2.11}$$

2. Para $|\psi_1\rangle$ e $|\psi_2\rangle$ arbitrários em V e $|\phi\rangle$ em W:

$$(|\psi_1\rangle + |\psi_2\rangle) \otimes |\phi\rangle = |\psi_1\rangle \otimes |\phi\rangle + |\psi_2\rangle \ \otimes |\phi\rangle \tag{2.12}$$

3. Para $|\psi\rangle$ arbritrário em V e $|\phi_1\rangle$ e $|\phi_2\rangle$ em W:

$$|\psi\rangle \otimes (|\phi_1\rangle + |\phi_2\rangle) = |\psi\rangle \otimes |\phi_1\rangle + |\psi\rangle \otimes |\phi_2\rangle \tag{2.13}$$

O produto tensorial entre dois vetores é:

$$|\psi\rangle \otimes |\phi\rangle = |\psi\rangle|\phi\rangle = |\psi\phi\rangle = \begin{bmatrix} \psi_1\phi_1 \\ \psi_1\phi_2 \\ \vdots \\ \psi_1\phi_n \\ \psi_2\phi_1 \\ \psi_2\phi_2 \\ \vdots \\ \psi_2\phi_n \\ \vdots \\ \psi_m\phi_1 \\ \psi_m\phi_2 \\ \vdots \\ \psi_m\phi_n \end{bmatrix} \tag{2.14}$$

onde $\psi_i\phi_j$ é o produto de números complexos.

Também é possível estender a definição de produto tensorial para matrizes. Seja A uma matriz de dimensão mxn, e B pxq:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}_{mp \times nq} \tag{2.15}$$

2.2 Postulados da mecânica quântica

A computação quântica é uma abordagem computacional da mecânica quântica. Através das leis da mecânica quântica pode-se codificar informação em um sistema quântico, processá-la usando portas quânticas e transmiti-la em canais quânticos.

Serão citados abaixo alguns postulados e critérios físico-matemáticos para efetuar computação com sistemas quânticos. Encontram se maiores explicações, ainda que básicas em Nielsen[NC00], e em Novaes[NS16] com maior rigor.

O primeiro postulado trata da descrição matemática de um sistema quântico isolado. O segundo trata da evolução dos sistemas físicos quânticos. O terceiro descreve a forma como pode-se extrair informações de um sistema quântico através de medições. O último postulado descreve a forma como sistemas quânticos diferentes podem ser combinados.

Postulado 1 (Espaço de estados) *Para todo sistema físico isolado, existe associado um espaço de Hilbert $\mathcal{H}$, chamado de* espaço de estados do sistema. *O sistema físico é totalmente descrito por seu vetor de estado, um vetor unitário $|\psi\rangle \in \mathcal{H}$.*

Este postulado estabelece que ao trabalhar com um sistema quântico, se está abstratamente trabalhando com vetores em um espaço vetorial. Os estados quânticos serão tratados pela sua representação vetorial no espaço de Hilbert e em sistemas finitos.

Uma das implicações diretas é que dado um espaço de Hilbert de dimensão finita n e uma base ortonormal $(|b_0\rangle, |b_1\rangle \cdots |b_{n-1}\rangle)$ de $\mathcal{H}$, tem-se um estado $|\psi\rangle$ que pode ser escrito como combinação linear:

$$|\psi\rangle = \sum_{i=0}^{n-1} \alpha_i |b_i\rangle \tag{2.16}$$

Além disso, como $|\psi\rangle$ é unitário, tem-se que:

$$\sum_{i=0}^{n-1} |\alpha_i|^2 = 1 \tag{2.17}$$

onde o valor α_i é conhecido como a *amplitude* do estado $|b_i\rangle$.

Postulado 2 (Evolução de estados) *A evolução de um sistema quântico fechado é descrita por uma transformação unitária. Se em um instante inicial t_1 o estado do sistema quântico é $|\psi\rangle$, e em um instante final t_2 o estado passa a ser $|\psi'\rangle$, então existe um operador unitário U, dependente somente de t_1 e t_2:*

$$|\psi'\rangle = U|\psi\rangle \tag{2.18}$$

Esta operação pode ser representada por uma matriz, e como a operação é uma tranformação unitária:

$$U^\dagger U = UU^\dagger = I \tag{2.19}$$

onde $U^\dagger$ é a matriz conjugada e transposta de U.

Uma consequência é que após aplicar um operador unitário sobre um vetor, sua norma se mantém. Portanto, o estado final também possuirá norma unitária como definido pelo Postulado 1. Outra consequência é que todo operador unitário é inversível, o que implica que as operações quânticas são *reversíveis*.

Postulado 3 (Medição dos estados) *As medições quânticas são descritas por operadores de medições $\{M_n\}$. São operadores que atuam sobre o espaço de estados do sistema. Os índices n se referem aos possíveis resultados da medição. Se o estado de um sistema quântico for $|\psi\rangle$ imediatamente antes da medição, a probabilidade de um resultado n ocorrer é dada por:*

$$p(n) = \langle\psi|M_n^\dagger M_n|\psi\rangle \tag{2.20}$$

O estado após a medição será:

$$|\psi'\rangle = \frac{M_n|\psi\rangle}{\sqrt{\langle\psi|M_n^\dagger M_n|\psi\rangle}} \tag{2.21}$$

Os operadores de medição devem obedecer à relação de completude:

$$\sum_n M_n^\dagger M_n = I \tag{2.22}$$

que garante que a soma das probabilidades seja 1,

$$\sum_n \langle\psi|M_n^\dagger M_n|\psi\rangle = \langle\psi| \left(\sum_n M_n^\dagger M_n\right) |\psi\rangle = 1 \tag{2.23}$$

Portanto, por este postulado, após realizar a medição, o estado quântico colapsa em um novo estado, alterando o sistema.

Postulado 4 (Composição dos estados) *O espaço de estado de um sistema quântico composto é o produto tensorial dos espaço de estados componentes. Ou seja, se tiver n sistemas quânticos numerados denotados por* 1 *até n, interagindo em um sistema quântico maior, o estado quântico total é denotado por:*

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle = |\psi_1\rangle|\psi_2\rangle \ldots |\psi_n\rangle \tag{2.24}$$

Denotando por $|\psi\rangle^{\otimes n}$ o produto tensorial do estado quântico $|\psi\rangle$ com ele mesmo (ex: $|\psi\rangle^{\otimes 3} = |\psi\rangle \otimes |\psi\rangle \otimes |\psi\rangle$), para $n = 1$, tem-se o próprio estado $|\psi\rangle$. Na base computacional quântica, assumindo que seja $|\psi\rangle = |0\rangle$, então, $|\psi\rangle \otimes |\psi\rangle = |0\rangle \otimes |0\rangle$ sendo denotado por $|00\rangle$ ou apenas por $|0\rangle$.

2.3 Superposição

A computação quântica, assim como a clássica, se utiliza de uma unidade fundamental para trabalhar, o q-bit. O q-bit possui um estado associado $|\psi\rangle$ que pode ser qualquer vetor unitário dentro do espaço vetorial bidimensional abrangido por $|0\rangle$ e $|1\rangle$ sobre os números complexos. O q-bit será apresentado com mais detalhes no próximo capítulo.

Definição 12 Superposição *é o fenômeno que permite um sistema quântico, não medido, estar em dois ou mais estados simultaneamente.*

Isto significa que um registrador quântico de n q-bits pode armazenar 2^n valores simultaneamente. Quando uma medida é realizada, o estado que antes estava em superposição colapsa para um único estado, e os demais se perdem. Portanto, pode-se dizer que um q-bit pode estar em superposição em tempo de execução, pois quando é realizada uma medida, o

valor deste se colapsa para um dos estados ($|0\rangle$ ou $|1\rangle$) com a probabilidade de cada um dos estados.

Diz-se que uma combinação linear $\sum_i \alpha_i|\psi_i\rangle$ é uma superposição de estados $|\psi_i\rangle$, com amplitude α_i para cada um deles. Por exemplo:

$$\frac{|\ 0\rangle - |1\rangle}{\sqrt{2}} \tag{2.25}$$

é uma superposição de $|0\rangle$ com amplitude $\frac{1}{\sqrt{2}}$ e $|1\rangle$ com amplitude $\frac{-1}{\sqrt{2}}$.

2.4 Emaranhamento

Definição 13 *O* emaranhamento *(ou entrelaçamento) quântico é o fenômeno que ocorre quando pares ou grupos de partículas interagem de forma que o estado quântico de cada uma não pode ser descrito independentemente, e ao invés disso, um estado quântico deve ser dado para o sistema como um todo.*

O entrelaçamento acontece quando duas partículas continuam se influenciando separadas, por qualquer distância. O que acontece em uma partícula é refletido na outra.

2.5 Teorema da não-clonagem

Não existe uma transformação que clona, ou que copia, sem prejuízo o estado de um q-bit qualquer para outro. Para clonar um estado é necessário realizar uma medição, entretanto, ao realizar a medição pode-se ter criado um q-bit clone, mas o q-bit original irá colapsar.

Teorema 1 (Não-clonagem) *Seja $\mathcal{H}$ um espaço de Hilbert, não existe uma transformação $U : \mathcal{H} \otimes \mathcal{H} \to \mathcal{H} \otimes \mathcal{H}$ tal que exista um $|s\rangle$, satisfazendo para qualquer $|\psi\rangle$:*

$$U\left(|\psi\rangle|s\rangle\right) = |\psi\rangle|\psi\rangle \tag{2.26}$$

Prova: Suponha que exista U, e que $|s\rangle$ é o estado que vai assumir as propriedades de $|\psi\rangle$ e $|\phi\rangle$, e sejam $|\psi\rangle$ e $|\phi\rangle$ q-bits arbitrários, tais que:

$$\begin{gathered} U\left(|\psi\rangle|s\rangle\right) = |\psi\rangle|\psi\rangle \\ \text{e} \\ U\left(|\phi\rangle|s\rangle\right) = |\phi\rangle|\phi\rangle \end{gathered} \tag{2.27}$$

como transformações unitárias conservam o produto interno e eles são multiplicativos sobre o produto tensorial, o produto interno entre as duas equações de 2.27 é:

$$\begin{gathered} \left(\langle\psi|\langle s|\right) U^\dagger U\left(|\phi\rangle|s\rangle\right) = \left(\langle\psi|\langle\psi|\rangle\right)\left(|\phi\rangle|\phi\rangle\right) \\ \langle s|s\rangle\langle\psi|\phi\rangle = \langle\psi|\phi\rangle\langle\psi|\phi\rangle \\ \langle s|s\rangle\langle\psi|\phi\rangle = \left(\langle\psi|\phi\rangle\right)^2 \end{gathered} \tag{2.28}$$

usando o fato de U ser unitário $\langle s|s\rangle = 1$:

$$\langle\psi|\phi\rangle = (\langle\psi|\phi\rangle)^2 \tag{2.29}$$

mas $x^2 = x$ só tem duas soluçoes, $x = 0$ ou $x = 1$, e portanto ou $|\psi\rangle = |\phi\rangle$ ou $|\psi\rangle$ e $|\phi\rangle$ são ortogonais. Então tal operação só clona estados ortogonais entre si, portanto a operação geral é impossível.

□

Os q-bits cujos estados quânticos são conhecidos são facilmente clonáveis, pois já estão colapsados, e qualquer medição realizada não irá mudar seu estado.

Capítulo 3

Computação quântica

Um computador quântico executa cálculos utilizando-se das propriedades da mecânica quântica, e isso já muda o paradigma em relação a computação clássica drasticamente.

Analogamente a um computador clássico, que funciona a partir de circuitos elétricos e portas lógicas manipulando bits, o computador quântico opera a partir de circuitos quânticos baseados em portas lógicas quânticas, manipulando a sua unidade fundamental, o q-bit.

3.1 Bit quântico (q-bit)

A estrutura básica de informação na computação clássica é o bit, a estrutura análoga na computação quântica é o bit quântico, que é usualmente chamado de *q-bit* (qbit, qubit ou quibit). Um q-bit é um estado quântico da forma:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.1}$$

*o*nde $\alpha, \beta \in \mathbb{C}$ (números complexos), as amplitudes, com $|\alpha|^2 + |\beta|^2 = 1$ e $\{|0\rangle, |1\rangle\}$ é uma base que expande $\mathcal{H}^2$. Essa base é chamada de base computacional e pode ser representada por notação matricial como segue:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3.2}$$

Quando sofre uma medição, o q-bit $|\psi\rangle$ pode colapsar para o estado $|0\rangle$ com probabilidade $|\alpha|^2$, ou para $|1\rangle$ com probabilidade $|\beta|^2$.

O vetor $|\psi\rangle$ é chamado de *superposição* dos vetores, com amplitudes α e β. Em mecânica quântica, o vetor é também chamado de *estado* [PLeNM12].

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{3.3}$$

A principal diferença, entre o bit clássico e o bit quântico, é que o bit clássico pode estar somente com um valor armazenado num determinado instante, esse valor é 0 ou 1. O bit

quântico (q-bit) está numa sobreposição de 0's e 1's num determinado instante, ou seja, 0 e 1 estão armazenados ao mesmo tempo. Realizar uma medição de um sistema quântico é um problema central na teoria quântica, e muitos estudos foram e continuam sendo feitos nessa área. Em um computador clássico, é possível a princípio saber sobre o estado de qualquer bit em memória, sem alterar o sistema. No computador quântico, a situação é diferente, q-bits podem estar em estados sobrepostos, ou até mesmo emaranhados, e o simples ato de medir um estado quântico altera seu estado.

3.1.1 Esfera de bloch

Como observado na equação 3.1, α e β são números complexos, então podem ser escritos como $\alpha = a + ib (a, b \in \mathbb{R})$ e $\beta = c + id (c, d \in \mathbb{R})$, então:

$$|\alpha|^2 + |\beta|^2 = a^2 + b^2 + c^2 + d^2 = 1 \tag{3.4}$$

A partir disso, o q-bit é interpretado como um vetor unitário $(a, b, c, d) \in \mathbb{R}^4$ e a esfera unitária de $\mathbb{R}^4$ como o lugar geométrico dos q-bits.

As amplitudes também podem ser expressas na forma $z = |z|e^{iArg(z)}$, onde $0 \leq Arg(z) \leq 2\pi$ é o argumento do número complexo z. Definindo $\gamma = Arg(\alpha)$ e $\phi = Arg(\beta) - Arg(\alpha)$, reescreve-se a equação 3.1 da forma:

$$|\psi\rangle = |\alpha|e^{i\gamma}|0\rangle + |\beta|e^{i(\gamma+\phi)}|1\rangle \tag{3.5}$$

Sendo $|\alpha| \geq 0, |\beta| \geq 0$ e $|\alpha|^2 + |\beta|^2 = 1$, define-se também ξ por meio das equações $cos(\xi) = |\alpha|$ e $sen(\xi) = |\beta|$ (onde $0 \leq \xi \leq \frac{\pi}{2}$). tomando $\theta = 2\xi, \theta \in [0, \pi]$, tem-se a forma polar de um q-bit:

$$|\psi\rangle = e^{i\gamma}\left[cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}sen\left(\frac{\theta}{2}\right)|1\rangle\right] \tag{3.6}$$

onde $\theta = 2arccos(|\alpha|) = 2arcsen(|\beta|)$ com $\theta \in [0, \pi]$, $\phi = Arg(\beta) - Arg(\alpha)$ com $\phi \in [0, 2\pi)$ e $\gamma = Arg(\alpha)$ com $\gamma \in [0, 2\pi)$.

Devido a algumas propriedades [CLM07], pode-se desprezar o fator $e^{i\gamma}$ (chamado fator de fase global), assim:

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}sen\left(\frac{\theta}{2}\right)|1\rangle \tag{3.7}$$

reescrevendo da seguinte forma:

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + cos(\phi)sen\left(\frac{\theta}{2}\right)|1\rangle + isen(\phi)sen\left(\frac{\theta}{2}\right)|1\rangle \tag{3.8}$$

Ou seja, o vetor $|\psi\rangle$ pode ser entendido como:

$$|\psi\rangle = \begin{bmatrix} a \\ c + id \end{bmatrix} \tag{3.9}$$

O espaço vetorial em questão tem dimensão quatro e uma de suas bases ortonormais é o conjunto:

$$\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} i \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ i \end{bmatrix}\} \tag{3.10}$$

Entretanto, é possível representar o vetor $|\psi\rangle$ utilizando apenas três vetores dessa bases:

$$|\psi\rangle = cos\left(\frac{\theta}{2}\right)\begin{bmatrix} 1 \\ 0 \end{bmatrix} + cos(\phi)sen\left(\frac{\theta}{2}\right)\begin{bmatrix} 0 \\ 1 \end{bmatrix} + sen(\phi)sen\left(\frac{\theta}{2}\right)\begin{bmatrix} 0 \\ i \end{bmatrix} \tag{3.11}$$

O subespaço gerado pelos elementos

$$\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ i \end{bmatrix}\} \tag{3.12}$$

tem dimensão três. Como esse subespaço está definido sobre o corpo dos reais, ele é isomorfo a $\mathbb{R}^3$. Pode-se então imaginar que, quando desprezado o fator de fase global de um q-bit, ele é "projetado" em um subconjunto de $\mathbb{R}^3$. Observa-se que o lugar geométrico determinado por $|\psi\rangle$é uma semiesfera de $\mathbb{R}^3$. Com algumas manipulações bem demonstradas em [CLM07], mostra-se que essa representação pode ser estendida para a Esfera de Bloch, como a figura 3.1.

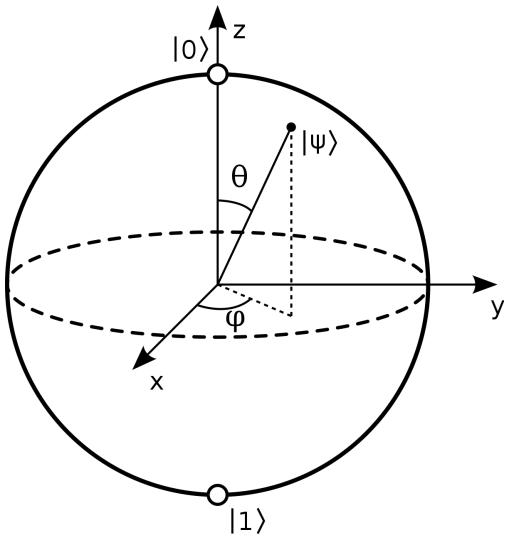

Figura 3.1: *Representação de um q-bit Esfera de Bloch [OS04].*

θ	ϕ	ψ	Observação
0	0	$\|0\rangle$	Polo Norte
π	0	$\|1\rangle$	Polo Sul
$\frac{\pi}{2}$	0	$\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}$	Equador - sobre eixo x
$\frac{\pi}{2}$	$\frac{\pi}{2}$	$\frac{\|0\rangle+i\|1\rangle}{\sqrt{2}}$	Equador - sobre eixo y

Tabela 3.1: *Direção e sentido do vetor representativo do q-bit para alguns estados.[OS04]*

Pode-se dizer que a Esfera de Bloch é o lugar geométrico dos vetores de Bloch. Considerando a forma polar de um q-bit, o Vetor de Bloch é dado por:

$$|\psi\rangle = \begin{bmatrix} cos(\phi)sen(\theta) \\ sen(\phi)sen(\theta) \\ cos(\theta) \end{bmatrix}, \phi \in [0, 2\pi), \theta \in [0, \pi] \tag{3.13}$$

Exemplo:

Encontrar o vetor que indica a localização do q-bit $|\psi\rangle = \frac{\sqrt{3}}{2}|0\rangle + \frac{1}{2}|1\rangle$ na Esfera de Bloch:

Passo 1 *Acha-se as amplitudes:*

$$\alpha = \frac{\sqrt{3}}{2} \text{ e } \beta = \frac{1}{2} \tag{3.14}$$

Passo 2 *Econtrando θ:*

$$\theta = 2arcos(\frac{\sqrt{3}}{2}) = 2arcsen(\frac{1}{2}) = \frac{\pi}{3} \tag{3.15}$$

Passo 3 *Econtrando ϕ:*

$$\phi = Arg(\frac{1}{2}) - Arg(\frac{\sqrt{3}}{2}) = 0 \tag{3.16}$$

Passo 4 *Substituindo no Vetor de Bloch:*

$$|\psi\rangle = \begin{bmatrix} cos(0)sen(\frac{\pi}{3}) \\ sen(0)sen(\frac{\pi}{3}) \\ cos(\frac{\pi}{3}) \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} \\ 0 \\ \frac{1}{2} \end{bmatrix} \tag{3.17}$$

3.2 Portas quânticas

Similar ao sistemas clássico de computação, as portas lógicas quânticas realizam a função de um operador lógico atuando na informação, ou nos q-bits, e assim manipulando ou alterando o seu estado

3.2.1 Portas quânticas de 1 q-bit

O conjunto de portas quânticas, matrizes de transformação, que realizam operações unitárias sobre um q-bit é infinito, pois as possibilidades de matrizes unitárias 2x2 são infinitas. As matrizes unitárias garantem que a computação possa ser reversível (dado um q-bit $|\psi_1\rangle$ em um estado arbitrário que passará pela porta quântica X produzindo o resultado $|\psi_2\rangle$ e a porta quântica inversa da X no q-bit $|\psi_2\rangle$, tem como resultado o q-bit inicial $|\psi_1\rangle$). Um vetor de estado (q-bit) deve ser unitário, e portanto, após a aplicação de uma porta quântica, tem-se outro vetor de estado que continuará sendo unitário [NC00].

Seguem algumas das portas mais usuais:

Porta Pauli-I

Esta porta também é conhecida como porta identidade e o resultado de sua operação não altera o estado do q-bit de entrada. Normalmente é omitida na maioria das referências. A porta Pauli-I é definida como

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.18}$$

A aplicação dela sobre um estado $|\psi\rangle = |0\rangle + |1\rangle$:

$$I|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |0\rangle + |1\rangle = |\psi\rangle \tag{3.19}$$

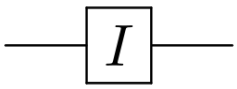

Figura 3.2: *Representação, em circuitos, da Porta Pauli-I.*

Porta NOT ou Pauli-X

A porta Pauli-X gira os q-bits π em torno do eixo x, como mostrado na figura 3.3. No caso em que o q-bit está posicionado em um dos estados clássicos, ela simplesmente atuará como um *"bit-flip"* semelhante a porta NOT, usado na computação clássica. É definida como:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{3.20}$$

Esta é uma matriz Hermitiana e satisfaz $X^2 = I$, onde I é a identidade. Os auto-vetores normalizados das matrizes de Pauli, quando transformados em vetores de Bloch, resultam em vetores pertencentes aos eixos. Assim, os auto-vetores normalizados do operador X são:

$$v_{x_1} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ e } v_{x_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{3.21}$$

e os vetores de Bloch associados a eles são:

$$B(v_{x_1}) = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \text{ e } B(v_{x_2}) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{3.22}$$

que pertencem ao eixo x da esfera.

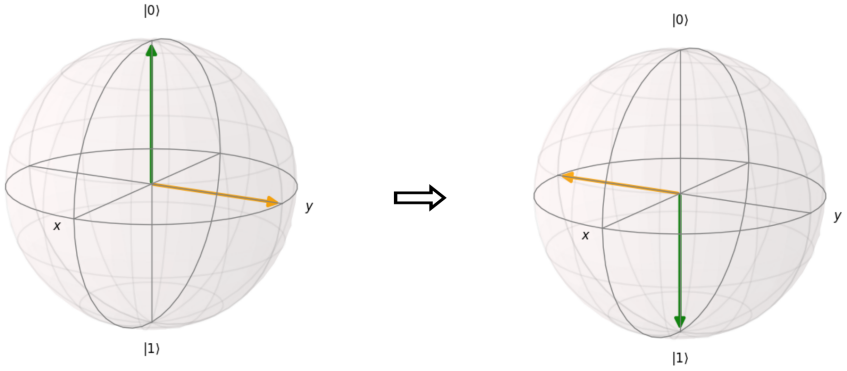

Figura 3.3: *O estado do q-bit antes e depois da porta Pauli-X [Gra18].*

A aplicação dela sobre os vetores $|0\rangle$ e $|1\rangle$:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \tag{3.23}$$

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle \tag{3.24}$$

Esta porta inverte o estado do q-bit de entrada.

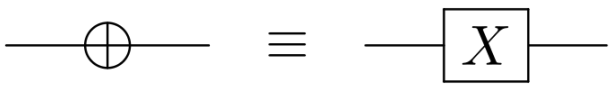

Figura 3.4: *Representações, em circuitos, da Porta Pauli-X.*

Porta Pauli-Y

O portão Pauli-Y gira o q-bit π radianos ao redor do eixo y. É definida como:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{3.25}$$

A aplicação dela sobre um estado $|\psi\rangle = |0\rangle + |1\rangle$:

$$Y|\psi\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} -i\alpha \\ i\beta \end{bmatrix} = i(|0\rangle - |1\rangle) \tag{3.26}$$

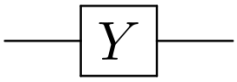

Figura 3.5: *Representação, em circuitos, da Porta Pauli-Y.*

Porta Pauli-Z

A porta Pauli-Z gira os q-bit π radianos em torno do eixo z. Este deixa a amplitude do q-bit inalterada, ou seja, $|0\rangle$ ainda será $|0\rangle$ e $|1\rangle$ ainda será $|1\rangle$, mas a fase do q-bit será deslocada por π radianos. A mudança de fase não afeta a probabilidade do estado clássico entrar em colapso, mas muda o estado do q-bit. Quando o q-bit está em superposição com uma probabilidade igual de colapso para $|0\rangle$ e $|1\rangle$, o resultado de um deslocamento de fase de π radianos será a transformação $|+\rangle \rightarrow |-\rangle$, $|-\rangle \rightarrow |+\rangle$. É definida como:

$$X = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{3.27}$$

A aplicação dela sobre um estado $|\psi\rangle = |0\rangle + |1\rangle$:

$$Z|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix} = |0\rangle - |1\rangle \tag{3.28}$$

Figura 3.6: *Representações, em circuitos, da Porta Pauli-Z.*

Porta Hadamard ou H

A porta de Hadamard é usada para forçar um q-bit a se sobrepor. A operação gira o q-bit π radianos em torno do eixo $x + z$ e pode ser visualizado na esfera de Bloch como uma rotação de $\frac{\pi}{2}$ radianos em torno do eixo y seguido de uma rotação de π radianos em torno do eixo x. Se a porta de Hadamard for aplicada a um q-bit em um estado clássico de $|0\rangle$ ou $|1\rangle$, a operação forçará o q-bit a uma sobreposição com amplitudes de probabilidade iguais de $|0\rangle$ e $|1\rangle$. É definida como:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3.29}$$

A aplicação dela sobre um estado, leva a uma superposição $|\psi\rangle = |0\rangle + |1\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{3.30}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{3.31}$$

Esta porta é utilizada para colocar o estado de um q-bit em superposição com mesma probabilidade para os dois estados. Ela é muito utilizada no circuito quântico que implementa a TFQ (Transformada de Fourier Quântica).

Figura 3.7: *Representação, em circuitos, da Porta Hadamard.*

Porta de Fase ou S

A porta de fase é definida como:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{3.32}$$

A aplicação dela sobre um estado genérico $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$S|\psi\rangle = S(\alpha|0\rangle + \beta|1\rangle) = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ 0 & i\beta \end{bmatrix} = \alpha|0\rangle + i\beta|1\rangle \tag{3.33}$$

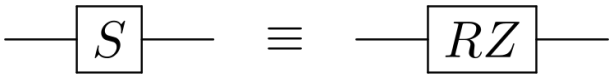

Figura 3.8: *Representação, em circuitos, da Porta de Fase*

Porta $\pi/8$ ou T

A Porta T é definida como:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{3.34}$$

A aplicação dela sobre um estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$T|\psi\rangle = \alpha|0\rangle + e^{i\pi/4}\beta|1\rangle) \tag{3.35}$$

Figura 3.9: *Representação, em circuitos, da Porta $\pi/8$ ou T.*

Porta R_k

A porta R_k é definida como:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{bmatrix} \tag{3.36}$$

A aplicação dela sobre um estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$:

$$R_k|\psi\rangle = \alpha|0\rangle + e^{2i\pi/2^k}\beta|1\rangle) \tag{3.37}$$

Matrizes de rotação

Das matrizes de Pauli, surge uma outra classe de matrizes, as *Matrizes de Rotação*, que são definidas como:

$$R_x(\theta) \equiv e^{-\frac{i\theta X}{2}} \text{ , } R_y(\theta) \equiv e^{-\frac{i\theta Y}{2}} \text{ e } R_z(\theta) \equiv e^{-\frac{i\theta Z}{2}} \tag{3.38}$$

Porém, para as matrizes A tais que $A^2 = I$, então:

$$e^{iAx} = \sum_{k=0}^{\infty} \frac{x^k(Ai)^k}{k!} = I + ixA - \frac{x^2 I}{2!} - \frac{ix^3 IA}{3!} + \frac{x^4 I}{4!} + \cdots = cos(x)I + iAsen(x) \tag{3.39}$$

Com esse resultado, é Possível reescrever as matrizes de rotação:

$$R_x(\theta) \equiv e^{-\frac{i\theta X}{2}} \equiv cos(\frac{\theta}{2})I + iXsen(\frac{\theta}{2}) \equiv \begin{bmatrix} cos(\frac{\theta}{2}) & -isen(\frac{\theta}{2}) \\ -isen(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{bmatrix} \tag{3.40}$$

$$R_y(\theta) \equiv e^{-\frac{i\theta Y}{2}} \equiv cos(\frac{\theta}{2})I + iYsen(\frac{\theta}{2}) \equiv \begin{bmatrix} cos(\frac{\theta}{2}) & sen(\frac{\theta}{2}) \\ sen(\frac{\theta}{2}) & cos(\frac{\theta}{2}) \end{bmatrix} \tag{3.41}$$

$$R_z(\theta) \equiv e^{-\frac{i\theta Z}{2}} \equiv cos(\frac{\theta}{2})I + iZsen(\frac{\theta}{2}) \equiv \begin{bmatrix} e^{-\frac{i\theta}{2}} & 0 \\ 0 & e^{\frac{i\theta}{2}} \end{bmatrix} \tag{3.42}$$

Estas matrizes representam rotações em torno dos eixos x, y e z na esfera unitária do $\mathbb{R}^3$.

3.2.2 Portas quânticas de múltiplos q-bits

Apesar de o conjunto de portas de um q-bit ser infinito, esse conjunto não é universal, isto é, não é suficiente para construir qualquer operação, sendo assim, para realizar operações sobre n q-bits é necessário utilizar portas com mais de um q-bit.

Porta CNOT quântica

A Porta CNOT atua em estados de 2 q-bits de entrada, o controle e o alvo. Uma porta controlada age de acordo com o q-bit de controle. Ela será ativada apenas quando o q-bit de controle estiver no estado $|1\rangle$. Os q-bits de controle e alvo podem ser estados superpostos, além disso, podem estar emaranhados.

A representação matricial da porta quântica CNOT é a dada por:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.43}$$

A ação pode ser representada de forma esquemática na base computacional da seguinte maneira:

$$CNOT|a, b\rangle \rightarrow |a, a \oplus b\rangle \tag{3.44}$$

onde $a, b \in \{0, 1\}$ e $\oplus$ é módulo 2.

$$\begin{aligned} CNOT|00\rangle &\rightarrow |00\rangle \\ CNOT|01\rangle &\rightarrow |01\rangle \\ CNOT|10\rangle &\rightarrow |11\rangle \\ CNOT|11\rangle &\rightarrow |10\rangle \end{aligned} \tag{3.45}$$

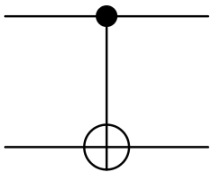

Figura 3.10: *Representação circuital da Porta CNOT Quântica. A linha superior representa o q-bit de controle, e a linha de baixo o q-bit-alvo*

Porta Toffoli quântica

O funcionamento da porta Toffoli é bastante semelhante a CNOT, também é uma porta controlada, só que com dois q-bits de controle. Seu funcionamento pode ser da seguinte

maneira, caso os q-bits $|a\rangle$ e $|b\rangle$ sejam iguais a $|1\rangle$ o q-bit $|c\rangle$ será negado.

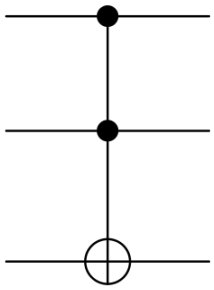

Figura 3.11: *Representação circuital da Porta Toffoli Quântica. As linhas superiores representam os q-bits de controle, e a linha de baixo o q-bit-alvo.*

A repreesntação matricial da porta Toffoli é dada por:

$$TOFFOLI = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.46}$$

A ação pode ser representada na base computacional da seguinte maneira:

$$TOFFOLI|a, b, c\rangle \rightarrow |a, b, c \oplus ab\rangle \tag{3.47}$$

onde $a, b, c \in \{0, 1\}$ e $\oplus$ é módulo 2. A base computacional possui 8 elementos nesse caso.

Porta SWAP

A porta de Swap, como pode ser visualizado na figura 3.12, é formado por três portas CNOT.

A evolução desta porta pode ser descrita como a troca de seus valores entre si.

A representação matricial da porta Swap é dada por:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.48}$$

A ação pode ser representada na base computacional da seguinte maneira:

$$SWAP|a, b\rangle \rightarrow |b, a\rangle \tag{3.49}$$

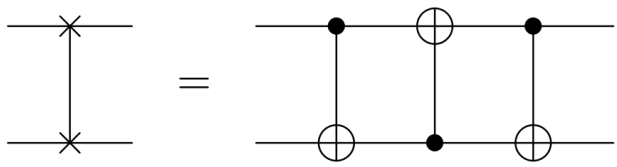

Figura 3.12: *Representação circuital da Porta Swap Quântica.*

Porta Fredkin

A porta Fredkin, que também é uma porta controlada, funciona com um q-bit de controle associado à uma porta Swap. Seu funcionamento pode ser da seguinte maneira, caso o q-bit de controle seja $|1\rangle$ e os q-bits alvo trocam de valor entre si.

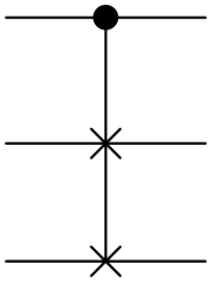

Figura 3.13: *Representação circuital da Porta Fredkin Quântica. A linha superiore representam o q-bit de controle, e as linha de baixo os q-bits-alvo.*

A repreesntação matricial da porta Fredkin é dada por:

$$FREDKIN = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.50}$$

Porta U_f ou oráculo

É uma porta muitas vezes chamada de "black box". É uma porta que não é pré-definida e, portanto, pode ser utilizada para manipular qualquer número de q-bits. Em outras palavras, ela é uma porta "genérica" onde se pode implementar qualquer operação, desde que esta obedeça às regras dos operadores unitários.

São operadores lineares unitários que calculam uma função característica arbitrária e desconhecida.

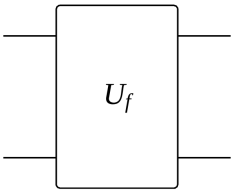

Figura 3.14: *Representação circuital de uma Porta Oráculo.*

Oráculos são amplamente utilizados em algoritmos quânticos voltados para problemas de busca ou extração de informações de funções desconhecidas.

Capítulo 4

Algoritmos e circuitos quânticos

Para um computador desenvolver determinada tarefa é necessário programá-lo, e para isso é preciso desenvolver um conjunto de procedimentos para realizar esta tarefa, a esse conjunto de procedimentos da-se o nome de *Algoritmo* [MF11].

Com o advento da computação quântica, os programas deverão ser construídos a partir de algoritmos quânticos. Neste ponto aparece um novo desafio, pois com esse novo paradigma, os futuros programadores deverão conhecer bem a forma como a informação deve ser tratada na perspectiva quântica [Gal07]. Os algoritmos quânticos nada mais são que aplicações de cirquitos quânticos.

4.1 Circuitos quânticos

Os circuitos quânticos são compostos por portas lógicas e "fios". Os fios não são necessariamente os objetos físicos costumeiros, eles apenas representam o transporte do q-bit através do circuito. O fio pode representar um fóton, ou outra partícula, se movendo de um local para outro no espaço. O circuito deve ser lido da esquerda para a direita [Meg08]. A notação gráfica é usada para descrever os processos computacionais utilizando os elementos descritos até aqui.

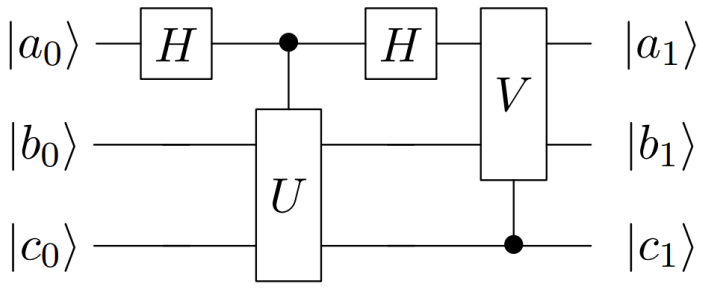

Figura 4.1: *Representação de um Circuito.*

Na Figura 4.1 $|a_0\rangle, |b_0\rangle$ e $|c_0\rangle$ representam os q-bits de entrada, as portas lógicas e as linhas horizontais que representam a evolução dos q-bits no tempo, e no final da linha $|a_1\rangle, |b_1\rangle$ e

$|c_1\rangle$ que são os q-bits finais ou resultado.

4.2 Paralelismo quântico

O Paralelismo é o que permite a execução de varias funções simultaneamente. Na computação convencional, esta técnica é implementada por meio de dois ou mais circuitos diferentes. Isto significa que para realizar duas ou mais operações simultaneamente, são necessários dois ou mais circuitos. Na computação quântica isso não é necessário, pois utilizando q-bits em estado de superposição é possível realizar a computação dos vários estados simultaneamente utilizando apenas um circuito [Sch09].

4.2.1 Algoritmo de Deutsch

Deutsch sugeriu que um computador quântico poderia realizar melhor seu potencial computacional se utilizasse o chamado **paralelismo quântico** [Deu85]. Desenvolveu um algoritmo que serve para exemplificar esse paralelismo.

Serão utilizados aqui algoritmos com algumas modificações da proposta original [NC00].

Problema: Dado um Oráculo U_f para uma função $f : \{0,1\} \rightarrow \{0,1\}$ descobrir se f é constante ($f(0) = f(1)$) ou balanceada ($f(0) \neq f(1)$).

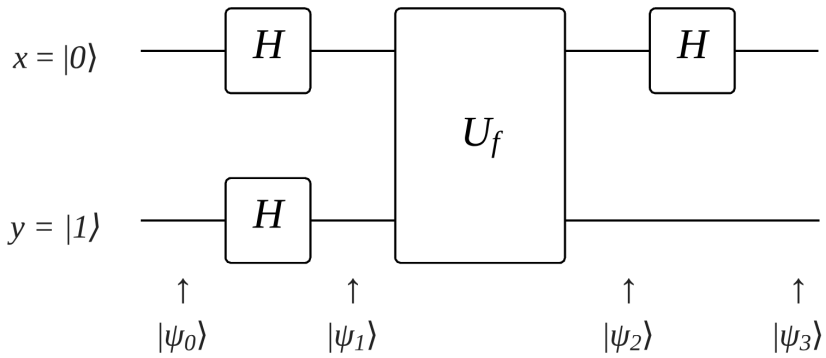

Figura 4.2: *Representação de um Circuito de Deutsch.*

Tem-se o estado de entrada $|\psi_0\rangle = |01\rangle$ que é enviado a duas portas de Hadamard, resulta:

$$|\psi_1\rangle = \left[\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{4.1}$$

Aplicando U_f em $|\psi_1\rangle$ obtém-se uma das duas possibilidades:

$$|\psi_2\rangle = \begin{cases} \pm\left[\frac{|0\rangle+|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{se} \quad f(0) = f(1) \\ \pm\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right]\left[\frac{|0\rangle-|1\rangle}{\sqrt{2}}\right] & \text{se} \quad f(0) \neq f(1) \end{cases} \tag{4.2}$$

Após as segunda porta de Hadamard atuar sobre o primeiro q-bit, tem-se:

$$|\psi_3\rangle = \begin{cases} \pm|0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{se} \quad f(0) = f(1) \\ \pm|1\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] & \text{se} \quad f(0) \neq f(1) \end{cases} \tag{4.3}$$

Observa-se que $f(0) \oplus f(1)$ será 0 se $f(0) = f(1)$ e 1 caso contrário, o resultado anterior pode ser reescrito da forma:

$$|\psi_3\rangle = \pm|f(0) \oplus f(1)\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \tag{4.4}$$

Realizando a medição sobre o primeiro q-bit é identificada se a função computada é balanceada (se o estado corresponder ao q-bit $|1\rangle$), ou constante (se o estado corresponder ao q-bit $|0\rangle$). Em um algoritmo clássico, seriam necessárias duas medições, uma para cada entrada.

4.3 Soma aritmética Vedral

As operações aritméticas básicas são fundamentais para muitos algoritmos. Uma primeira proposta de algoritmo de soma foi dada por Vedral [VBE95].

Este algoritmo é composto de portas CNOT e Toffoli, que são utilizados em dois operadores, *Carry* e *Sum*.

O operador Carry tem como função avaliar três q-bits (a_0, b_0, c_0) e Identificar se o q-bit de alvo (c_1) deve ser colocado em $|1\rangle$ caso mais de um q-bit de entrada tenham valor $|1\rangle$.

A porta SUM tem como função avaliar três q-bits (a_0, b_0, c_0) e colocar o resultado da soma módulo 2 em b_0.

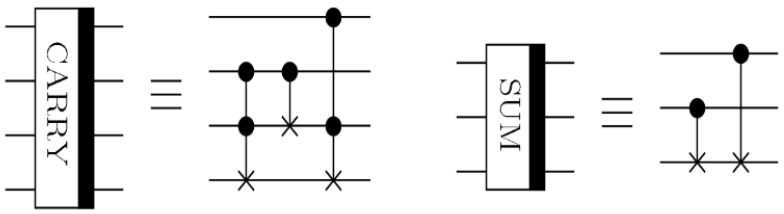

Figura 4.3: *Representação dos circuitos de Carry e Sum [VBE95].*

Este algoritmo de soma é uma sequência de portas Carry seguidas de uma sequência de portas SUM, conforme mostrado na figura 4.3:

Problema: Sejam $a, b \in \mathbb{N}$, em base binária e cada um com $n + 1$ q-bits, efetuar a sua soma binária.

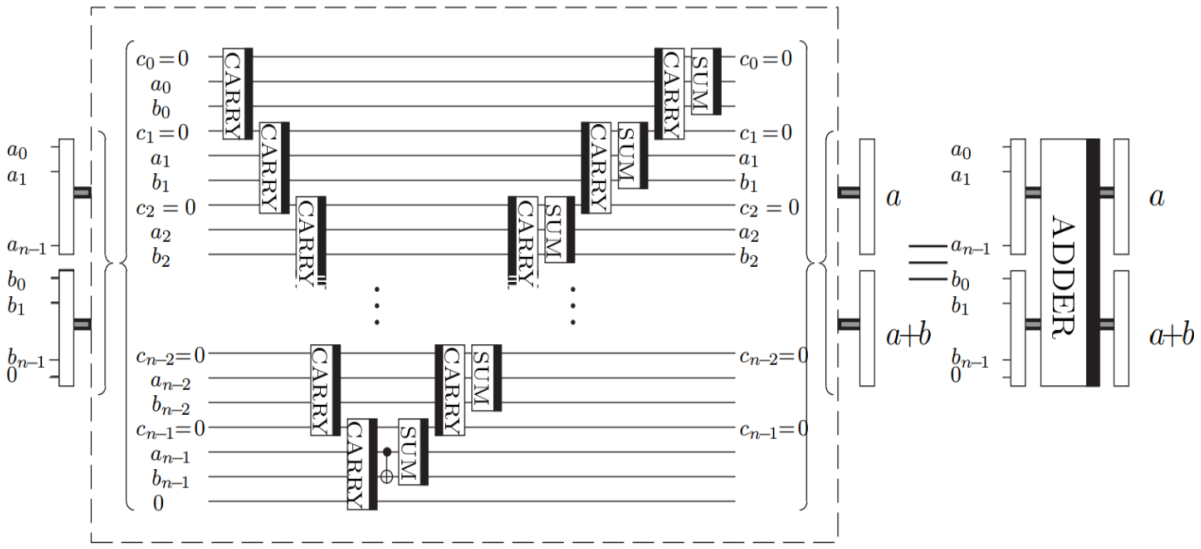

Figura 4.4: *Representação do circuito de Soma de Vedral [VBE95], adaptado por Kowada [Kow06].*

O circuito de soma opera sobre dois estados quânticos $|a\rangle$ e $|b\rangle$. Sendo cada um deles:

$$\begin{aligned} |a\rangle &= |a_n, a_{n-1}, \cdots, a_0\rangle \\ |b\rangle &= |b_n, b_{n-1}, \cdots, b_0\rangle \end{aligned} \tag{4.5}$$

Assim a soma:

$$|a\rangle + |b\rangle = |a, a+b\rangle = |a_n, a_{n-1}, \cdots, a_0, s_{n+1}, s_n, \cdots, s_1, s_0\rangle \tag{4.6}$$

Onde $s_{n+1}, s_n, \cdots, s_1, s_0$ são os q-bits resultantes da soma aritmética. São necessários n q-bits de a, n de b e $n+1$ auxiliares, totalizando $3n+1$ q-bits.

Observação: Para realizar a *Subtração*, basta reverter o circuito de soma.

4.3.1 Exemplo de soma Vedral no circuito

Problema: Dado dois números $a, b \in \mathbb{N}$, tal que $a = 1$ e $b = 2$:

$$\begin{aligned} |a\rangle &= |a_1, a_0\rangle \text{ com } a_1 = 0 \text{ e } a_0 = 1 \\ |b\rangle &= |b_1, b_0\rangle \text{ com } b_1 = 1 \text{ e } b_0 = 0 \end{aligned} \tag{4.7}$$

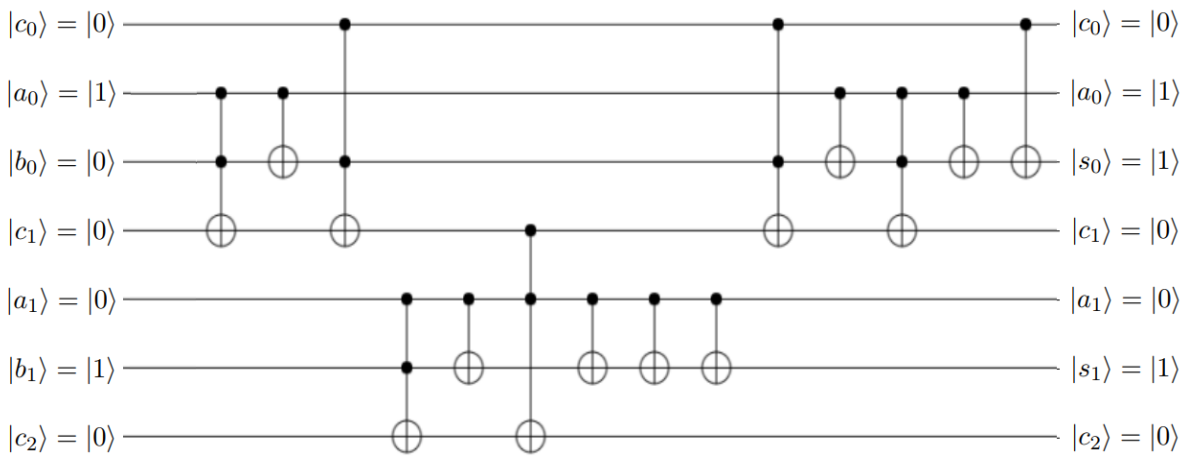

Figura 4.5: *Representação de um Circuito de soma Vedral para doi bits.*

Resultando $|a\rangle + |b\rangle = |s_1, s_0\rangle = |11\rangle$. Ou seja 11_2 na base binária, que é 3_{10} na base decimal.

4.4 Transformada de Fourier quântica

A Transformada de Fourier é uma das ferramentas mais úteis da matemática na ciência e tecnologia modernas, sendo usada em vários campos de aplicação. A transformada é especialmente útil quando se tem algo com certa periodicidade, uma vez que nos ajuda a extrair tal periodicidade da função para analisar os dados.

A Transformada de Fourier discreta atua sobre conjuntos de dados discretos. Supondo um vetor de números complexos $(x_0, \cdots, x_{N-1})$, cujo comprimento N é um parâmetro fixo, a transformada irá gerar um vetor, também complexo, $(y_0, \cdots, y_{N-1})$ de forma que:

$$y_k \equiv \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}} \tag{4.8}$$

Como os q-bits são vetores de números complexos, a equação 4.8 pode ser reescrita como uma *Transformada de Fourier Quântica* (TFQ) [GN95], apenas mudando a notação. Em uma base ortonormal $|0\rangle, \cdots, |N-1\rangle$, sua transformação sobre um estado arbitrário é dada por:

$$\sum_{j=0}^{N-1} x_j |j\rangle \rightarrow \sum_{k=0}^{N-1} \left(\sum_{j=0}^{N-1} x_j e^{\frac{2\pi i j k}{N}} \right) |k\rangle \tag{4.9}$$

Definindo $N = 2^n$, onde n é o número de q-bits usados no modelo quântico com uma base computacional $|0\rangle, |1\rangle, \cdots, |N-1\rangle$. É possível escrever j usando a representação binária $j = j_1 j_2 \cdots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \cdots + j_n 2^0$. Tem-se a TFQ na forma conhecida como representação de produto:

$$|j_1 \cdots j_n\rangle \rightarrow \frac{\left(|0\rangle + e^{\frac{2\pi i 0.j_n}{2^1}}|1\rangle\right)\left(|0\rangle + e^{\frac{2\pi i 0.j_{n-1}j_n}{2^1}}|1\rangle\right)\cdots\left(|0\rangle + e^{\frac{2\pi i 0.j_1 j_2 \cdots j_n}{2^1}}|1\rangle\right)}{2^{\frac{n}{2}}} \tag{4.10}$$

A representação do produto 4.10 facilita a construção do circuito, como representado na figura 4.6. Essa equivalência é bem explicada em Nielsen [NC00].

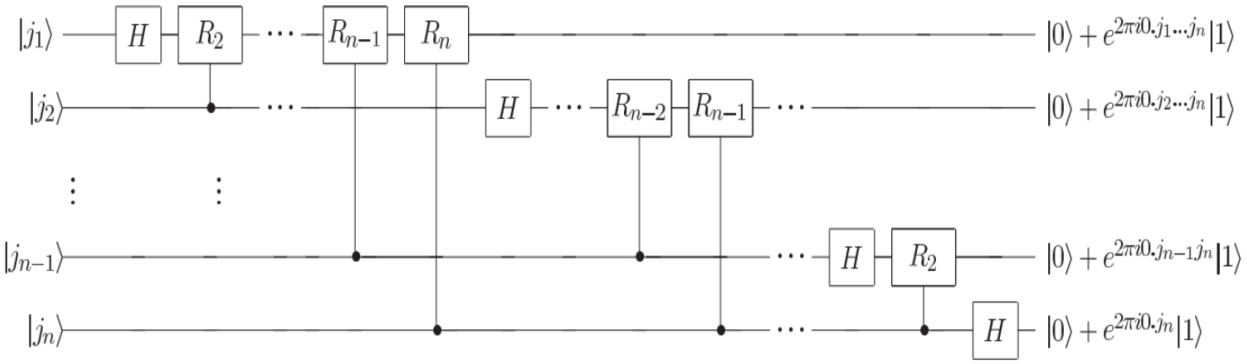

Figura 4.6: *Representação do circuito de Transformada de Fourier Quântica [NC00].*

A TFQ pode ser representada como um matriz:

$$TFQ_{2^n} = \frac{1}{\sqrt{2^n}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{2^n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(2^n-1)} \\ 1 & \omega^3 & \omega^6 & \cdots & \omega^{3(2^n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{2^n-1} & \omega^{2(2^n-1)} & \cdots & \omega^{(2^n-1)(2^n-1)} \end{bmatrix} \tag{4.11}$$

onde $\omega = e^{\frac{2\pi i}{2^n}}$.

4.5 Soma aritmética Draper

O circuito quântico de soma proposto por Draper [Dra00] difere significativamente do Vedral. O algoritmo de soma Vedral implementa uma técnica que utiliza o paradigma convencional para realizar a operação. De acordo com Draper, um somador quântico não pode ser construído de forma similar a um somador convencional.

A ideia básica de Draper é utilizar a *Transformada de Fourier Quântica*. Considerando a soma de dois valores a e b, primeiro é computado $f(a)$, que é a TFQ de a e usa o valor b para encontrar o resultado $f(a+b)$. Por fim, aplica-se a TFQ inversa para obter o resultado de $a+b$.

Na figura 4.7, $1, 2, \cdots, n$ representam as portas quânticas de rotação condicional onde R_k é o ângulo a ser rotacionado.

Este circuito utiliza como entrada dois estados quânticos, o estado $|a\rangle = |a_n, a_{n-1}, \cdots, a_0\rangle$ e o estado $|b\rangle = |b_n, b_{n-1}, \cdots, b_0\rangle$, de n q-bits cada. Na figura 4.7 não aparece, mas o estado $|a\rangle$ passa primeiro por uma TFQ e é transformado no estado $\phi|a\rangle = |\phi_n(a), \phi_{n-1}(a), \cdots, \phi_1(a)\rangle$, para então entrar no circuito somador.

Como se pode observar, o que difere este circuito da TFQ são as rotações condicionadas aos q-bits b_i, externos aos q-bits transformados [KP07].

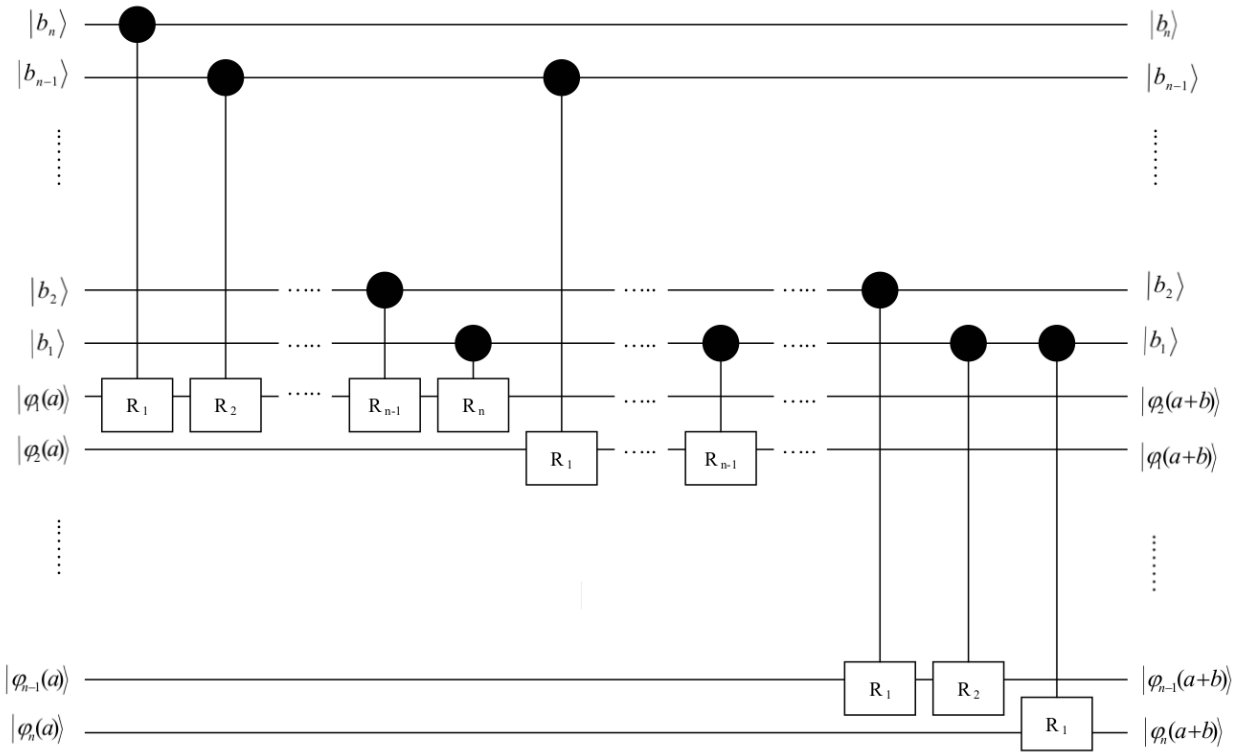

Figura 4.7: *Representação do circuito de Soma Draper [Dra00].*

Ao final da aplicação de todas as rotações no q-bit $|\phi_n(a)\rangle$ tem-se o resultado $|\phi_n(a+b)\rangle$. Após a transformação em todos os q-bits a, chega-se ao resultado final $|\phi_n(a+b)\rangle|\phi_{n-1}(a+b)\rangle \cdots |\phi_1(a+b)\rangle$.

Neste circuito, para implementar uma soma de dois números com n q-bits cada, são necessários $2n$ q-bits.

4.5.1 Exemplo de soma Draper

Problema: Dado dois números $a, b \in \mathbb{N}$, tal que $a = 1$ e $b = 2$:

A TFQ para dois q-bits:

$$TFQ_4 \equiv \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} \tag{4.12}$$

Passo 1 *Tranformar a e b em números binários:*

$$\begin{gathered} a = 1_{10} = 01_2 \\ \text{e} \\ b = 2_{10} = 10_2 \end{gathered} \tag{4.13}$$

Passo 2 *Calcular o produto tensorial de a_0 com a_1:*

$$|0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.14}$$

Passo 3 *Aplicar a TFQ_4:*

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ i \\ -1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ i \end{bmatrix} \tag{4.15}$$

Passo 4 *Aplicar o produto tensorial de b_1 com a_0:*

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 0 \\ 1 \\ i \end{bmatrix} \tag{4.16}$$

Passo 5 *Aplicar a porta controlada R_1 em a_1 condicionada à b_0:*

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 1 \\ i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ 0 \\ 1 \\ -i \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -i \end{bmatrix} \tag{4.17}$$

Passo 6 *Aplicar o produto tensorial de b_1 com a_1:*

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -i \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -i \\ 0 \\ 0 \end{bmatrix} \tag{4.18}$$

Passo 7 *Aplicar a porta controlada R_2 em a_1 condicionada à b_1:*

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&i&0\\0&0&0&i\end{bmatrix}\begin{bmatrix}1\\-i\\0\\0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\\0\\0\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\otimes\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix} \tag{4.19}$$

Passo 8 *Aplicar o produto tensorial de b_1 com a_0:*

$$\begin{bmatrix}1\\0\end{bmatrix}\otimes\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix} \tag{4.20}$$

Passo 9 *Aplicar a porta controlada R_1 em a_0 condicionada à b_1:*

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&i\end{bmatrix}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}=\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\\0\\0\end{bmatrix}=\begin{bmatrix}1\\0\end{bmatrix}\otimes\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} \tag{4.21}$$

Passo 10 *Aplicar o produto tensorial de a_0 com a_1:*

$$\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix}\otimes\frac{1}{\sqrt{2}}\begin{bmatrix}1\\-i\end{bmatrix}=\frac{1}{2}\begin{bmatrix}1\\-i\\-1\\i\end{bmatrix} \tag{4.22}$$

Passo 11 *Aplicar a TFQ_4^T:*

$$\frac{1}{4}\begin{bmatrix}1&1&1&1\\1&-i&-1&i\\1&-1&1&-1\\1&i&-1&-i\end{bmatrix}\begin{bmatrix}1\\-i\\-1\\i\end{bmatrix}=\frac{1}{4}\begin{bmatrix}0\\0\\0\\4\end{bmatrix}=\begin{bmatrix}0\\0\\0\\1\end{bmatrix}=\begin{bmatrix}0\\1\end{bmatrix}\otimes\begin{bmatrix}0\\1\end{bmatrix} \tag{4.23}$$

Passo 12 *Transformando em decimal:*

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle \otimes |1\rangle = 11_2 = 3_{10} \tag{4.24}$$

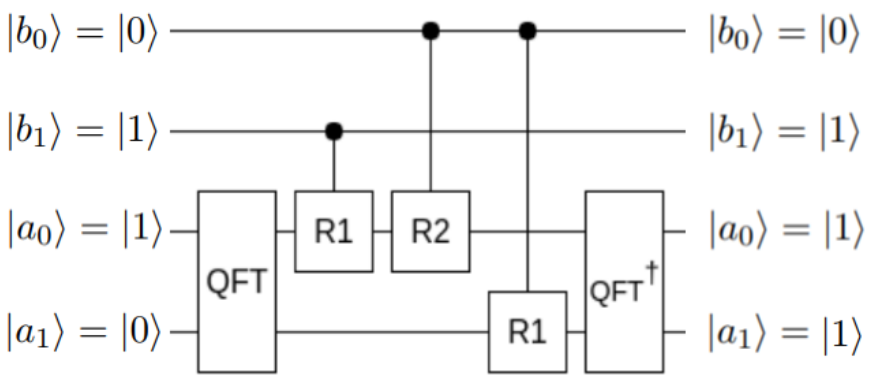

Figura 4.8: *Representação do circuito de Soma Draper para dois bits (1+2).*

4.6 Outros algoritmos quânticos

Os computadores quânticos são projetados para superar os computadores clássicos, entretanto, executando algoritmos quânticos. As áreas nas quais eles podem ser aplicados incluem criptografia, busca, otimização, simulação de sistemas quânticos e resolução de grandes sistemas de equações lineares. Alguns dos mais importantes, que além de demonstrarem a sua capacidade de cálculo, formam uma base para outros algoritimos, segundo Montanaro [Mon15], são:

- **Deutsch-Jozsa** é uma generalização do algoritmo de Deutsch. Este permite determinar se uma função é constante ou balanceada, mas desta vez a função possui múltiplos valores de entrada;

- **Shor** é fundamental para demonstrar o poder e a importância da computação quântica. Este pode ser usado para fatorar números primos, o que significa que ele pode ser usado para quebrar códigos de criptografia, quando um computador quântico prático for construído. Este algoritmo chamou a atenção de muitas pessoas;

- **Grover** pode ser descrito como um algoritmo de busca de banco de dados quântico. O algoritmo de Grover demonstra o poder de um computador quântico em que o algoritmo reduz significativamente o número de operações necessárias para resolver o problema, em comparação com um computador clássico. Suas principais aplicações são na área de identificação de padrões, bioinformática, conectividade em grafos, encontrar o mínimo em uma lista não classificada de inteiros, etc;

- **Algoritmos de caminhada quântica** que permitem projetar novos algoritmos quânticos mais eficientes e rápidos;

- **Algoritmos de simulação quântica** que permitem simular comportamentos e propriedades quânticas como a equação de schrödinger e teletransporte;
- **Harrow** resolve sistemas de equações lineares. O algoritmo estima o resultado de uma medida escalar no vetor solução para um dado sistema linear de equações.

Além dos citados, estão tantos outros entrando nas áres de *machine learning*, inteligência artificial, tomografia quântica e comportamento humano [RP10].

Capítulo 5

Simuladores de circuitos quânticos

As simulações desempenham um papel importante em diversas áreas de conhecimento humano, seja no estudo ou no desenvolvimento delas. Para a computação quântica, se tornou uma das alternativas mais viáveis para o estudo e o desenvolvimento da área.

Trabalhos relacionados ao desenvolvimento de simuladores têm produzido ferramentas, tais como simuladores de circuitos quânticos e linguagens de programação, os quais facilitam a compreensão de algum aspecto relacionado à computação quântica.

Um simulador de circuitos quânticos permite descrever (textualmente e/ou graficamente) um algoritmo em termos de portas e circuitos e testar esse algoritmo para um estado quântico específico através da simulação do hardware assim descrito. A linguagem de circuitos quânticos tem descrito os principais algoritmos quânticos conhecidos, ela é mais próxima dos físicos e dos engenheiros eletricistas, pois possui bastante similaridade com o seu análogo clássico que é amplamente conhecido.

Como a computação quântica é interdisciplinar, ou seja, envolve conhecimentos de física, matemática e computação, é salutar fornecer ferramentas que descrevam este paradigma de forma interessante para todas estas áreas.

Deve-se salientar que qualquer abordagem de simulação do paradigma computacional quântico em sistemas clássicos irá sofrer limitações. Ainda assim, a disponibilidade de um sistema computacional que permita uma descrição em nível apropriado de um algoritmo quântico e uma "máquina" para executar (ou simular) o algoritmo, facilitam tanto o ensino quanto o próprio desenvolvimento de algoritmos.

Nielsen [NC00] sugere, que para projetar bons algoritmos, deve-se "desligar" da intuição clássica, parcialmente, e usar efeitos verdadeiramente quânticos.

5.1 Alguns simuladores

Gustavo Cabral [Cab04] divide os Simuladores quânticos em dois tipos, os simbólicos e os universais. Os simbólicos são aqueles em que se desenvolve os algoritmos algebricamente. Enquanto os universais são aqueles que utilizam portas lógicas quânticas, em um circuito

quântico.

Os simuladores escolhidos foram os universais pela usabilidade e didática. Algoritmos quânticos são implementados, principalmente, utilizando a ideia de circuitos [NC00]. Além disso, nesse trabalho, os simuladores serão divididos offline, e online.

5.1.1 Simuladores *offline*

Foram selecionados dois simuladores que podem ser utilizados diretamente em sistemas, sem estar conectado à uma rede de computadores. Um simulador para dispositivos móveis (celular) e outro para computadores pessoais comuns.

QCS - Quantum Circuit Simulator

Desenvolvido como um projeto de curso de graduação por Mert Çikla, como requisito para formação no curso de Bacharelado em Ciência da Computação pela *Izmir University* (Turquia). O aplicativo teve sua última versão disponibilizada para Android em 2013 [APK].

O QCS é um aplicativo simples que permite simular o comportamento de portas quânticas básicas em celulares ou emuladores. Com ele é possível simular, em até seis q-bits, o comportamento das portas Pauli-X, Pauli-Y, Pauli-Z, Hadamard, CNOT e Swap. Após rodar a simulação, o aplicativo, retorna as probabilidades de cada resultado. O aplicativo é intuitivo e utiliza um sistema de *drag-and-drop* para a montagem dos circuitos, ou seja, basta o usuário arrastar a porta lógica de interesse para uma das linhas horizontais para a montagem do circuito.

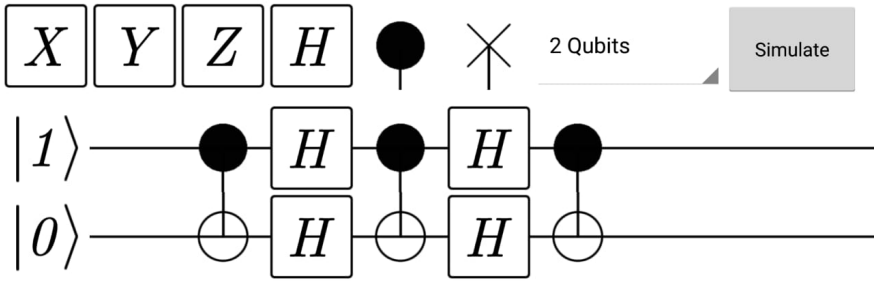

Figura 5.1: *Imagem da interface do aplicativo QCS, com um circuito de SWAP [Pla].*

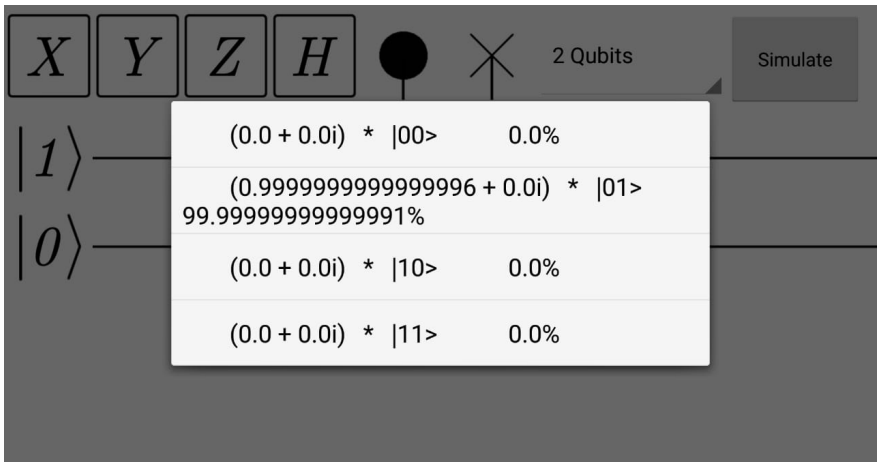

Figura 5.2: *Imagem da interface do aplicativo QCS, com sa solução do circuito SWAP [Pla].*

No o exemplo anterior é construído de um circuito quântico equivalente a uma porta Swap.

Simulador Zeno

O simulador Zeno foi desenvolvido em 2004, em Java, como trabalho de dissertação de mestrado de Gustavo Eulálio Cabral, pela Universidade de Campina Grande [Cab04]. Apesar da última versão ser de 2006, é uma ferramenta didática completa [dEeCeIQI].

Um pouco mais completo que o QCS, é um programa para computadores pessoais. Conta com três tipos de saída: ket, matriz densidade e histograma.

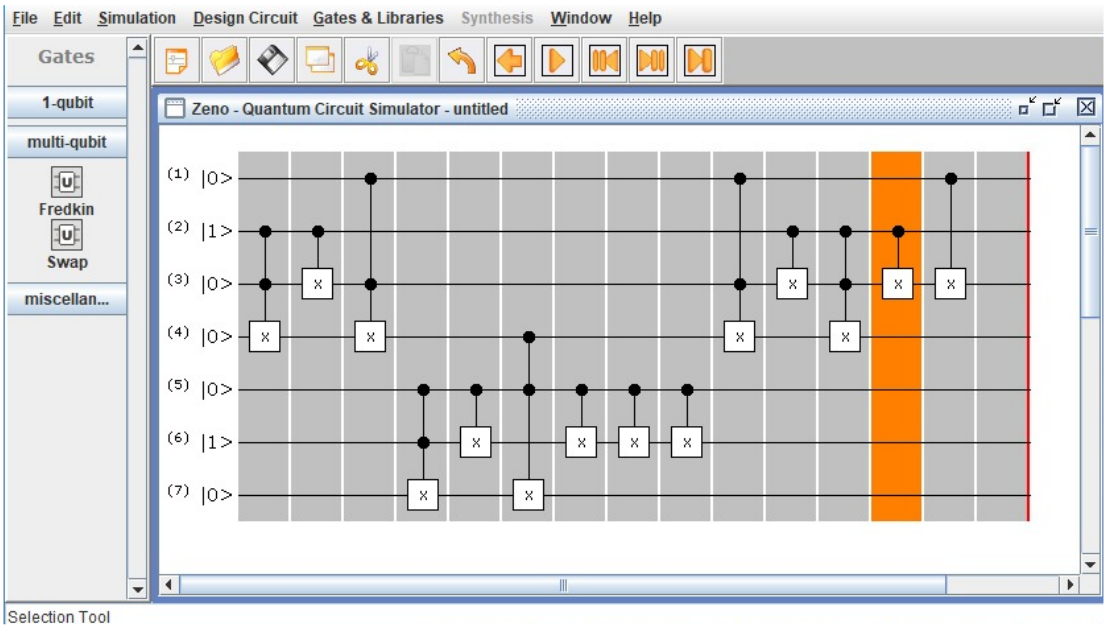

Figura 5.3: *Imagem da interface do Simulador Zeno, com um circuito somador Vedral para 3 q-bits, 1+2.*

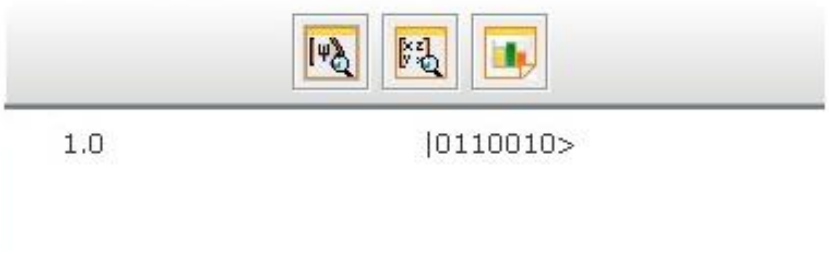

Figura 5.4: *Tela do resultado, tipo ket, da soma Vedral.*

O circuito apresenta leitura do resultado invertida, em relação a forma apresentada no capítulo anterior, porém sem comprometê-lo.

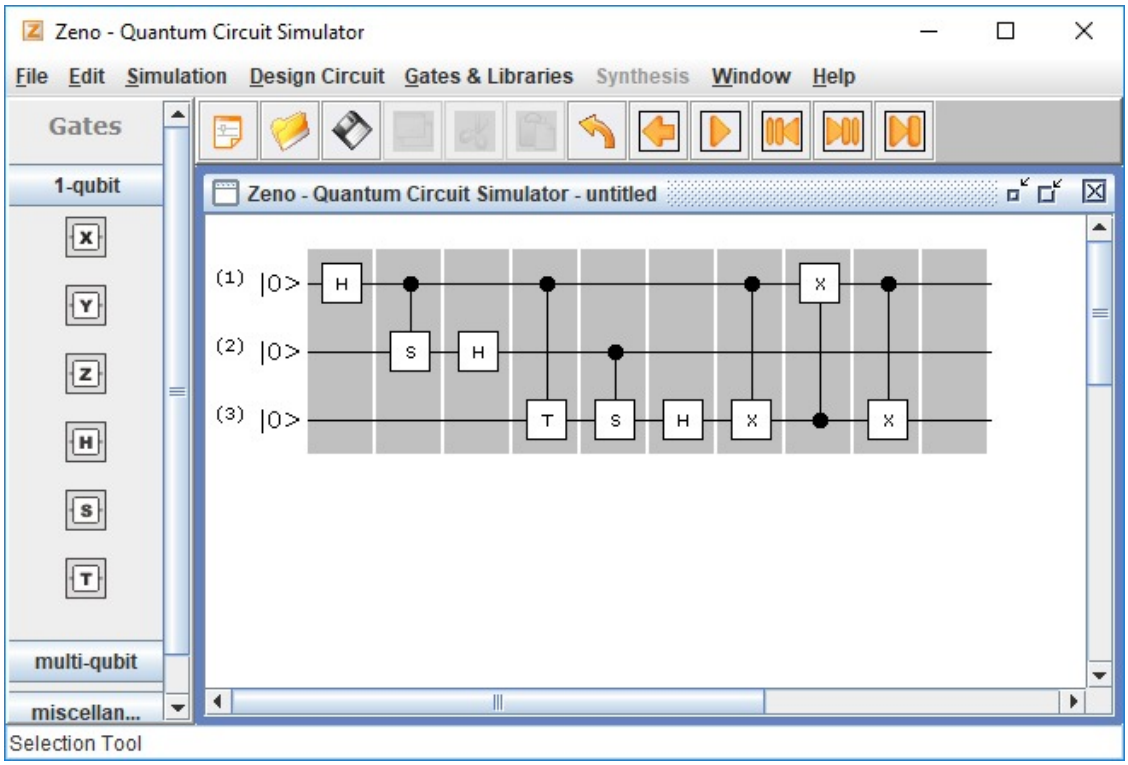

Figura 5.5: *Imagem da interface do Simulador Zeno, com um circuito de TFQ para 3 q-bits.*

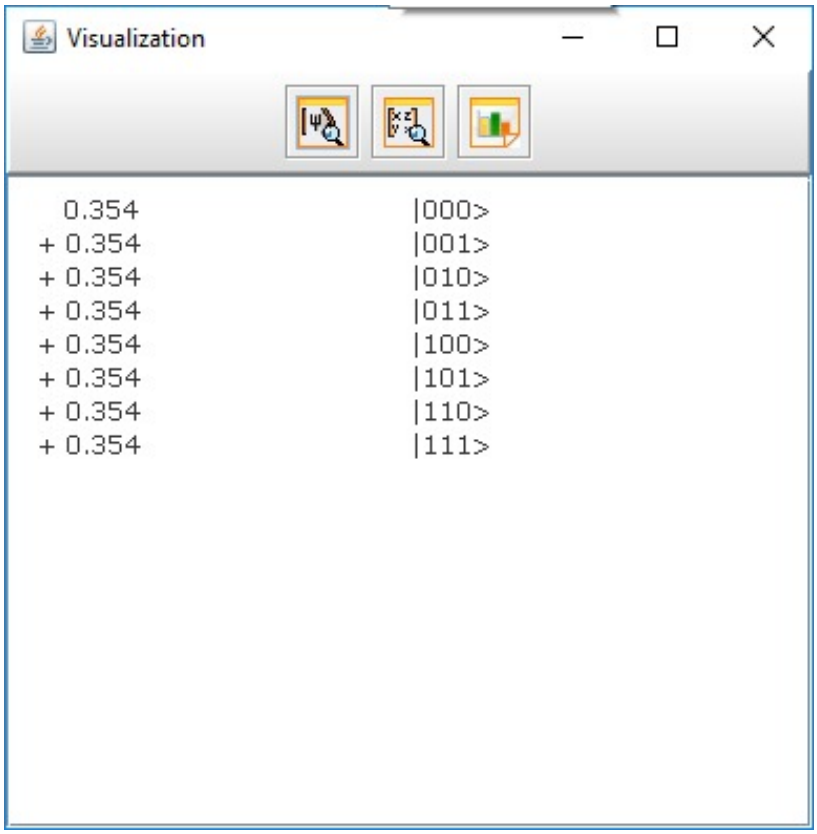

Figura 5.6: *Tela do resultado, tipo ket, do Zeno para TFQ.*

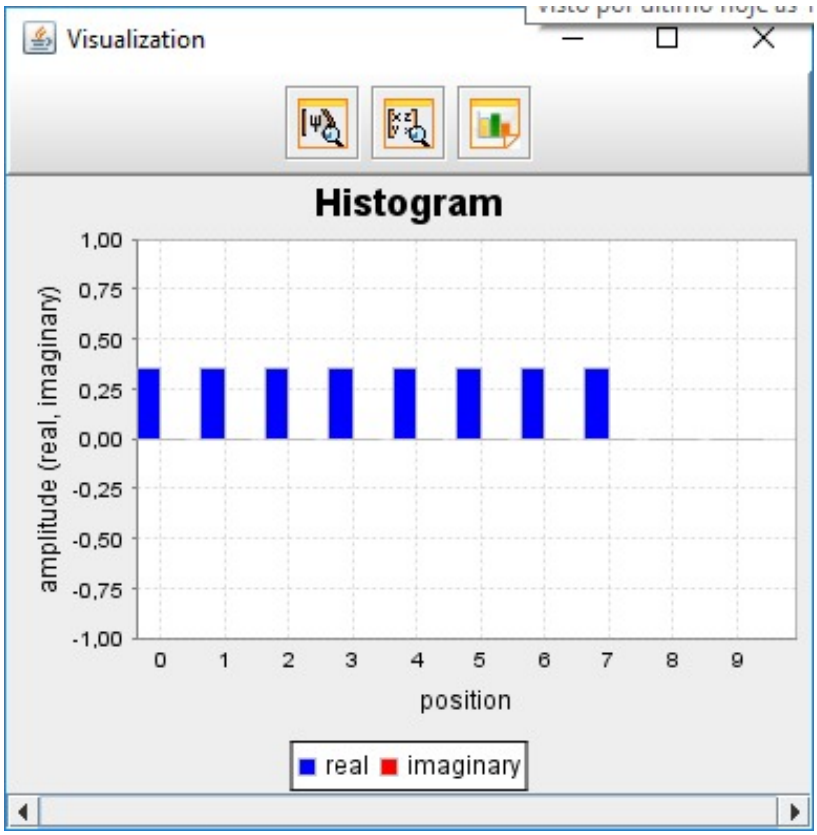

Figura 5.7: *Tela do resultado, tipo histograma, do Zeno para TFQ [Cab04].*

As telas de resultado, das figuras 5.6 e 5.7, mostram que para um conjunto de q-bits $|000\rangle$ que passa pela TFQ tem como resultados as combinações possíveis com amplitude dos reais 0.354.

5.1.2 Simuladores *online*

Foram selecionados dois simuladores que podem ser utilizados em qualquer computador pessoal ou dispositivo portátil com um navegador padrão (chrome, firefox, etc) e conexão à rede mundial de computadores, não requerendo assim instalação prévia de programas.

IBM Q Experience

IBM-Q é um processador quântico com 5 q-bits que pode ser acessado remotamente via internet através de uma plataforma de acesso manipulada pelo navegador, o IBM Q Experience, que permite manuseio *draganddrop* ou via linha de comando [Qb].

A plataforma permite simular o circuito nos servidores clássicos (disponibilizando até 20 q-bits), ou rodar no dispositivo real (computador quântico).

O programa disponibiliza as principais portas: Pauli-I, Pauli-X, Pauli-Y, Pauli-Z, Hadamard, Fase, T, CNOT, além de um medidor e uma barreira de evolução e de portas editáveis, onde pode ser escolhar a rotação desejada. É possível, também, montar o circuito utilizando linha de comando com a linguagem QASM.

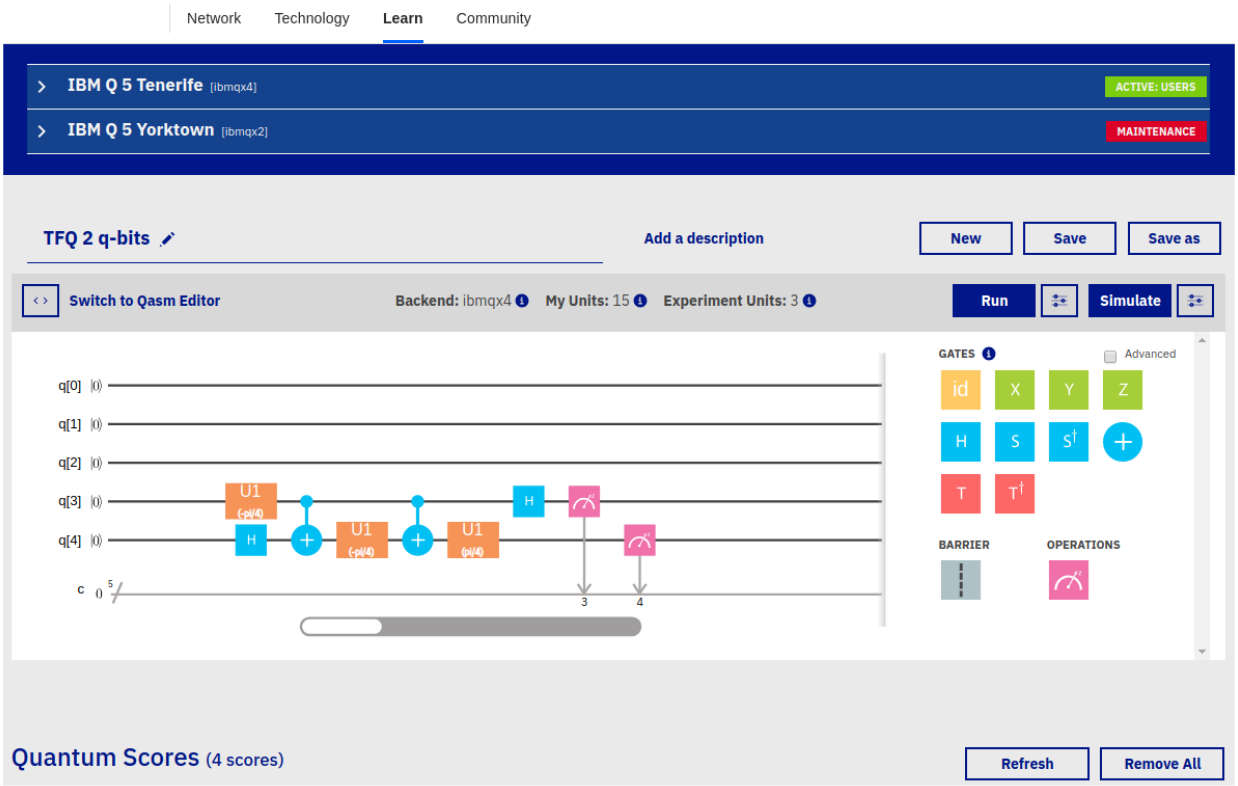

Figura 5.8: *Imagem da interface do IBM-Q, com um circuito de TFQ para 2 q-bits [Qa].*

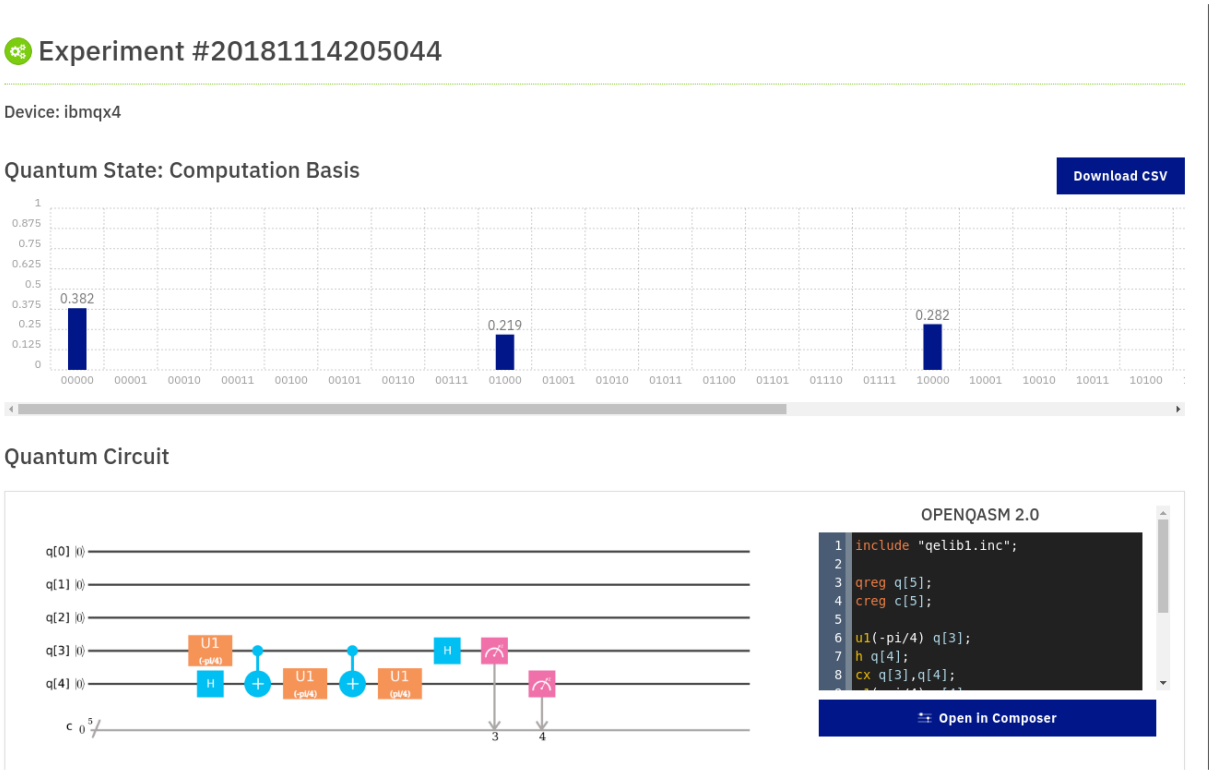

Figura 5.9: *Imagem da interface do IBM-Q, com o resultado da TFQ para 2 q-bits [Qa].*

O resultado obtido na figura 5.9 foi obtido não pelo simulador, mas pelo dispositivo real.

Simulador *QUIRK*

Talvez o simulador universal mais completo disponível gratuitamente. Desenvolvido por Craig Gidney, apesar de trabalhar na *Google Quantum Computing* em Santa Barbara, construiu o programa nas horas vagas [Gida].

O programa conta com diversos exemplos e recursos para construir circuitos complexos, e já disponibiliza alguns circuitos em forma de portas lógicas, como é o caso da QFT, que aplica a Transformada de Fourier Quântica para qualquer quantidade de q-bits. É possível também montar portas matricialmente e salvá-las para uso posterior no circuito [Gidb].

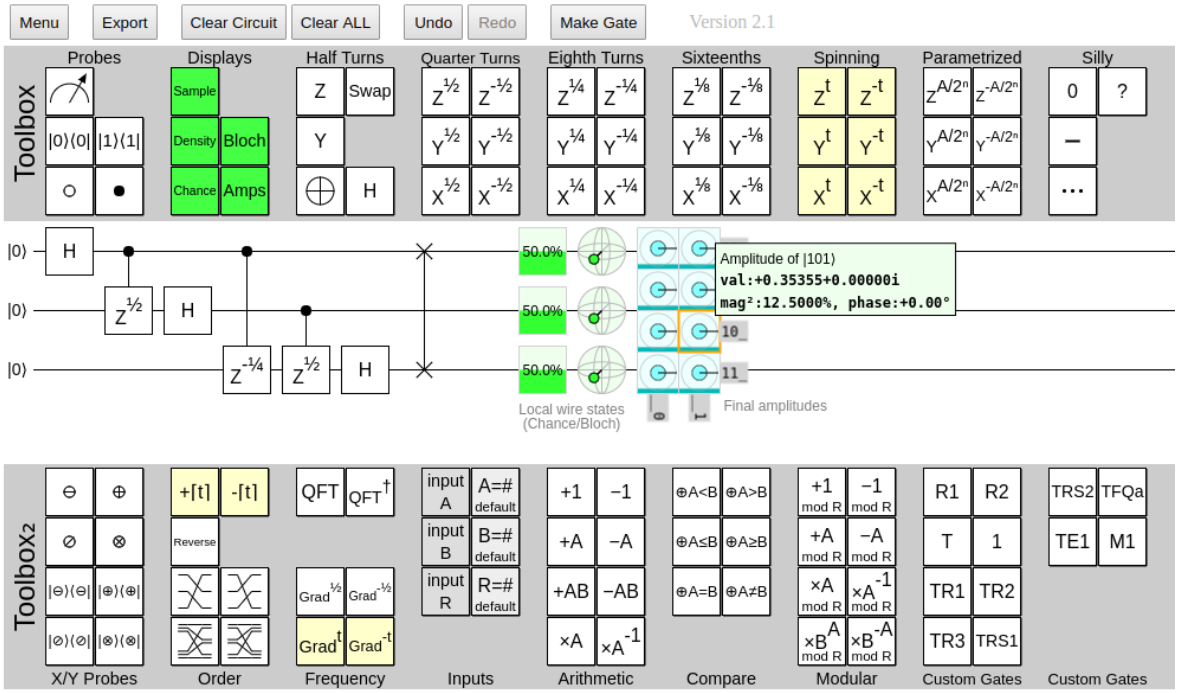

Figura 5.10: *Imagem da interface do Quirk, TFQ para 3 a-bits, com entrada* $|000\rangle$.

O resultado da TFQ na figura 5.10 está no meio da imagem, com cada possibilidade, com suas probabilidades, e, como mostrado na figura tem 0.35455 de amplitude nos reais, conferindo com o resultado da imagem 5.6.

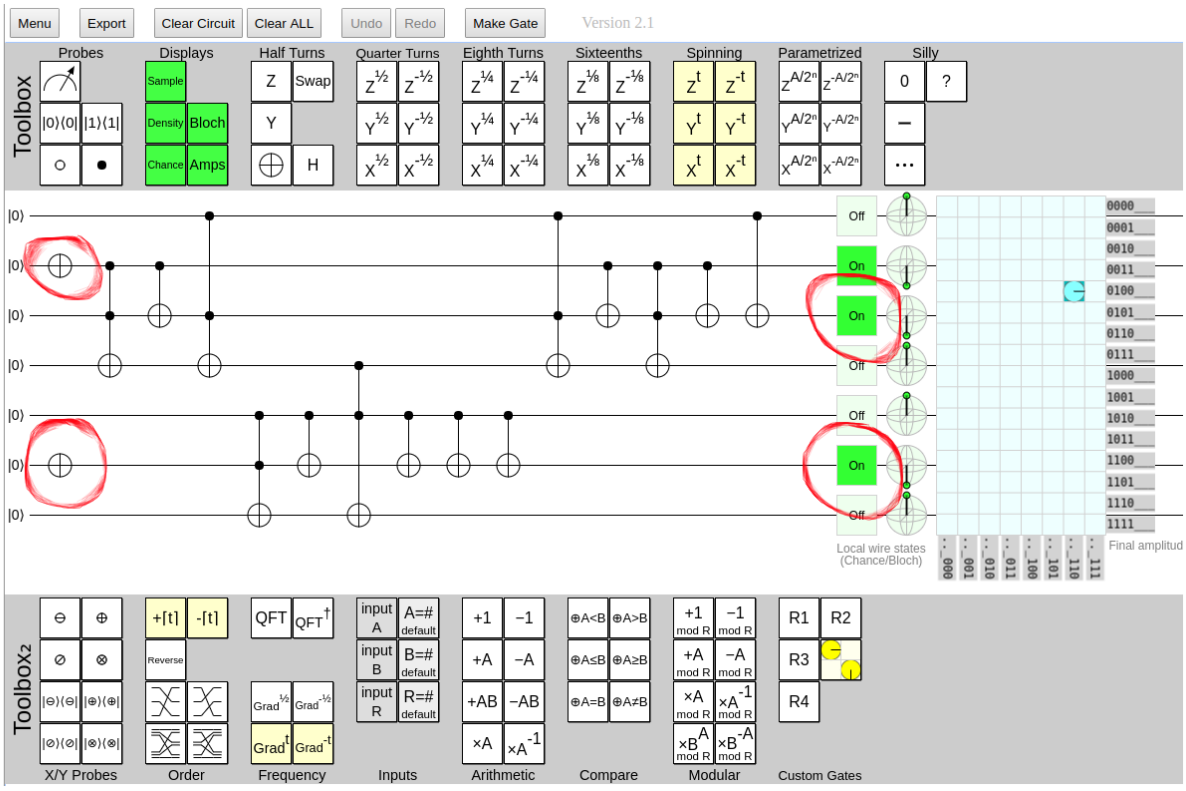

Figura 5.11: *Imagem da interface do Quirk com soma vedral para 2 q-bits, 1+2.*

Na figura 5.11, foi necessário colocar duas portas Pauli-X, pois não é possível colocar valores diferentes de zero nas entradas. No final do circuito estão destacados duas chaves, são as saídas efetivas da resposta, representam 11_2 ou 3_{10}.

5.2 Linguagens de programação

Existem dois grupos principais de linguagens de programação quântica: imperativas e funcionais. Na abordagem imperativa, há um código que descreve detalhadamente os passos que o computador deve seguir para alcançar determinado objetivo. A abordagem funcional envolve compor o problema em uma série de funções a serem executadas.

5.2.1 Imperativas

QCL

QCL (*Quantum Computing Language*), criada por Bernhard Ömer em 1998, é a primeira linguagem real para programação quântica, imperativa. Sua sintaxe se assemelha à de linguagens de programação clássicas e estruturadas, como a linguagem C ou Pascal. Ela permite a simulação de computadores quânticos em máquinas comuns, sendo a mais indicada para o ensino da linguagem de programação de computador quântico [Öm02].

Exemplo: Transformada de Fourier Quântica, de n q-bits em QCL

```
1 // transformada de fourier para 3 q-bits
2
3 operator dft(qureg q) {           // main
4   const n= #q;                    // define n tamanho da entrada
5   int i; int j;                   // declara contadores de loops
6   for i=1 to n {
7     for j=1 to i-1 {              // aplicar portais de fase condicional
8       V(pi/2^(i-j),q[n-i] & q[n-j]);
9     }
10    H(q[n-i]);                    // rotacoo de q-bit
11  }
12  flip(q);                        // troca a ordem dos q-bits da saida
13 }
```

QASM

QASM (*Quantum Assembly Language*), é uma representação intermediária para instruções quânticas. A linguagem foi descrita pela primeira vez em julho de 2017 [CBSG17] e o código-fonte foi lançado como parte do QISKit (Kit Quantum da IBM), para uso com sua

plataforma de computação quântica da IBM Q Experience. A linguagem tem qualidades semelhantes às linguagens tradicionais de descrição de hardware.

Exemplo: Transformada de Fourier Quântica, de 3 q-bits em QASM

```
1 # transformada de fourier para 3 q-bits
2
3   def c-S,1,'S'   # define as portas controladas
4   def c-T,1,'T'
5
6   qubit j0        # define os q-bits
7   qubit j1
8   qubit j2
9
10  h j0            # hadamard no primeiro q-bit
11  c-S j1,j0       # S primeiro q-bit com controle no segundo
12  c-T j2,j0       # T primeiro q-bit com controle no terceiro
13  h j1            # hadamard no segundo q-bit
14  c-S j2,j1       # S segundo q-bit com controle no terceiro
15  h j2            # hadamard no terceiro q-bit
16  swap  j0,j2     # SWAP entre o primeiro e o terceiro q-bit
```

5.2.2 Funcionais

QPL

QPL (*Quantum Programming Language*), criada por Peter Selinger em 2004, é a primeira linguagem funcional quântica. Esta linguagem é estaticamente digitada e permite detectar erros em tempo de compilação em vez de tempo de execução [Sel04].

QML

QML (*Quantum Markup Language*), criada por Altenkirch e Grattage em 2005, é uma linguagem funcional baseada na XML (Extensible Markup Language). Possui controle e dados quânticos [AG05].

QHaskell

QHaskell (*Quantum Haskell*), criada por Juliana K. Vizzotto e Antonio C. R. Costa, da Universidade Federal do Rio Grande do Sul, lançada em 2006, é uma linguagem funcional mista. Possui tanto controle e dados quânticos quanto clássicos [VdRC06].

Capítulo 6

Conclusões

Este trabalho teve como principal escopo contribuir a compreensão a divulgação desse campo transversal às áreas da matemática, computação, física e engenharia, que é a computação quântica.

A adoção do paradigma quântico na computação parece ser uma trajetória natural, e caminha concomitante com a diminuição do tamanho dos dispositivos eletrônicos presentes nos computadores, como já previa a Lei de Moore. Seria um erro pensar nela como mais uma dentre muitas tentativas de substituição de uma tecnologia em vias de esgotamento.

O embasamento teórico necessário transita entre a matemática, em especial a álgebra linear no espaço dos complexos, e conceitos de física moderna. Essas compreensões são indispensáveis ao interessado na área.

Os circuitos quânticos são a forma mais simples para entender do funcionamento de computadores quânticos. A impossibilidade de construção de computadores quânticos em grandes escalas faz com que simulação deles em computadores clássicos seja valorosa, já que para desenvolver novos algoritmos quânticos é necessário também testá-los. Simuladores não devem apenas fornecer o resultado de cálculos, devem permitir a extração de informações importantes acerca de algoritmos simulados. Portanto, um bom simulador pode ser uma excelente ferramenta ao pesquisador.

6.1 Considerações finais

Finalizando a segunda década do século XXI, a computação quântica e seus conceitos ainda são relativamente restritos e pouco divulgados. Espera-se que este texto contribua para que outros se interessem por esse campo da ciência.

6.2 Sugestões para pesquisas futuras

Não apenas a teoria, mas também as aplicações e a divulgação que incentivam o autor a continuar nesta área fascinante. O estudo de novos algoritmos quânticos, bem como a sua

implementação, visando diversas áreas do conhecimento. Uma área, também interessante, seria a construção de novos simuladores.

Apêndice A

Números complexos

Para maior aprofundamento no assunto, é sugestão a obra de Soares [Soa09].

A.1 Soma e multiplicação

Definição 1 *Um corpo é um conjunto* $\mathbb{C}$ *em que pode definir duas operações:*

$$\begin{aligned} + : \mathbb{C} \times \mathbb{C} &\to \mathbb{C} \\ (a, b) &\mapsto a + b \end{aligned} \tag{A.1}$$

$$\begin{aligned} \cdot : \mathbb{C} \times \mathbb{C} &\to \mathbb{C} \\ (a, b) &\mapsto a \cdot b = ab \end{aligned} \tag{A.2}$$

tais que para todos $a, b, c \in \mathbb{C}$ *valem:*

1. (Associatividade) $a + (b + c) = (a + b) + c$ e $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
2. (Comutatividade) $a + b = b + a$ e $a \cdot b = b \cdot a$;
3. (Existência de elemento neutro) existem elementos distintos $0 \in \mathbb{C}$ e $1 \in \mathbb{C}$ tais que $a + 0 = a$ e $a \cdot 1 = a$;
4. (Existência de inversos) Para todo $a \in \mathbb{C}$ Existe $-a \in \mathbb{C}$ tal que $a + (-a) = 0$ e se $a \neq 0$ existe $a - 1 \in \mathbb{C}$ tal que $a \cdot a - 1 = 1$;
5. (Distributividade) $a \cdot (b + c) = a \cdot b + a \cdot c$.

Definição 2 *Um número complexo é uma expressão do tipo:* $z = x + iy$*, em que* x *e* y *são números reais e* i*, chamado unidade imaginária, satisfaz a propriedade* $i^2 = -1$*. O número* $x = Re(z)$ *é a parte real de* z *e* $y = Im(z)$ *é a parte imaginária de* z*.*

Para definir a soma e a multiplicação de números complexos usa-se as operações de soma e multiplicação de números reais e considera cada número complexo como um polinômio em

i, de modo que a soma de dois números complexos $z_1 = x_1 + iy_1$ e $z_2 = x_2 + iy_2$ é dada por:

$$z_1 + z_2 = (x_1 + x_2) + i(y_1 + y_2), \tag{A.3}$$

e o produto:

$$z_1 z_2 = x_1 x_2 + i x_1 y_2 + i x_2 y_1 + i^2 y_1 y_2 = (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1). \tag{A.4}$$

Definição 3 *O conjugado de um número complexo $z = x + iy$ é o número complexo $z^* = x - iy$. A norma de z é $\mid z \mid = \sqrt{z \cdot z^*} = \sqrt{x^2 + y^2}$. Um número complexo z é chamado unitário se $\mid z \mid = 1$.*

A.2 Representação geométrica

Pode-se representar os números complexos geometricamente usando o plano cartesiano. O número complexo $z = x + iy$ é representado pelo ponto (x, y) no plano cartesiano e $\mid z \mid$ representa a distância euclidiana entre os pontos $(0, 0)$ e (x, y). A partir da representação geométrica verifica-se ver que se $r = \mid z \mid$ e φ é o ângulo formado entre a reta que liga os pontos (x, y) e $(0, 0)$ e o eixo x então:

$$z = r(cos(\varphi) + isen(\varphi)) \tag{A.5}$$

Desse modo, se z é um complexo unitário então $z = cos(\varphi) + isen(\varphi)$ para algum $\varphi \in \mathbb{R}$.

A.3 Exponencial complexa

Algumas funções definidas para números reais podem ser facilmente generalizadas para $\mathbb{C}$. Entre elas está a função exponencial.

Definição 4 *A exponencial de um número complexo z é definida por*

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!} \tag{A.6}$$

A exponencial está bem definida para todo número complexo. Isso segue do fato:

$$\sum_{n=0}^{\infty} \frac{\mid z^n \mid}{n!} = \sum_{n=0}^{\infty} \frac{\mid z \mid^n}{n!} = e^{|z|} \tag{A.7}$$

em Soares [Soa09] é possível admirar a proposição de convergência de de séries de números complexos, além de observar a validade das seguintes propiedades:

1. $e^{z+w} = e^z \cdot e^w$, para todos $z, w \in \mathbb{C}$;

2. $e^{-z} = \frac{1}{e^z}$;

3. $e^0 = 1$;

4. $(e^z)^n = e^n z$, para todo $z \in \mathbb{C}$ e $n \in \mathbb{Z}$;

5. $e^z \neq 0$.

A.4 Argumento de um complexo

Definição 5 *Considerando $c = a + bi$, sendo $a; b \in \mathbb{R}$. O argumento de c é representado por* ***Arg(z)*** *ou θ e é determinado por:*

$$Arg\,(z) = \theta \Leftrightarrow \begin{cases} cos(\theta) \\ sen(\theta) \\ 0 \leq \theta \leq 2\pi \end{cases} \tag{A.8}$$

Para p número $c = 0$ não é definido argumento. A condição afirma que para cada complexo c equivale apenas um argumento θ.

Referências Bibliográficas

[ABC11] Bárbara Amaral, Alexandre T. Baraviera e Marcelo O. T. Cunha. *Mecânica Quântica para Matemáticos em Formação*. 28º Colóquio Brasileiro de Matemática - IMPA, 1º edição, 2011. 7

[AG05] Thorsten Altenkirch e Jonathan Grattage. A functional quantum programming language. *arXiv:quant-ph/0409065v5*, 2005. 52

[APK] APK. Quantum circuit simulator. https://www.apkmonk.com/app/mert.qcs/. último acesso em 30/09/2018. 44

[BGK18] Sergey Bravyi, David Gosset e Robert König. Quantum advantage with shallow circuits. *Science*, 2018. 6

[Cab04] Gustavo E. M. Cabral. Uma ferramenta para projeto e simulação de ciruitos quântios. Dissertação de Mestrado, Centro de Ciênias e Tenologia da Universidade Federal de Campina Grande, Brasil, 2004. x, 43, 45, 47

[CBSG17] Andrew W. Cross, Lev S. Bishop, John A. Smolin e Jay M. Gambetta. Open quantum assembly language. *arXiv:1707.03429v2*, 2017. 51

[CG17] Andrés Cassinello e José Luiz S. Gómes. *O Mistério Quântico*. Crítica, 1º edição, 2017. 7

[CLM07] Luiz M. Carvalho, Carlile C. Lavor e Valeria S. Motta. Caracterização matemática e visualização da esfera de bloch. *TEMA Tend. Mat. Apl. Comput. SBMac*, 2007. 18, 19

[dEeCeIQI] Instituto de Estudos em Computação e Informação Quânticas IQUANTA. Zeno - quantum circuit simulator. http://www.dsc.ufcg.edu.br/~iquanta/zeno/. último acesso em 30/09/2018. 45

[Deu85] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, 1985. 5, 32

[Dra00] Thomas G. Draper. Addition on a quantum computer. *arXiv:quant-ph/0008033*, 2000. ix, 36, 37

[Fay82] Richard P. Faynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982. 5

[Gal07] Ernesto F. Galvão. *O que é Computação Quântica*. Vieira e Lent, 1º edição, 2007. 31

[Gida] Craig Gidney. Quirk quantum circuit simulator. http://algassert.com/2016/05/22/quirk.html. último acesso em 05/10/2018. 49

[Gidb] Craig Gidney. Quirk quantum circuit simulator. http://algassert.com/quirk/. último acesso em 05/10/2018. 49

[GN95] Robert B. Griffiths e Chi-Sheng Niu. Semiclassical fourier transform for quantum computation. *arXiv:quant-ph/9511007*, 1995. 35

[Gra18] Marcus Granström. Performance evaluation of quantum fourier transform on a modern quantum device. *School of Electrical Engineering and Computer Science - Stockholm*, 2018. ix, 22

[Gro96] Lov K. Grover. Fast quantum mechanical algorithm for database search. *28th Annual ACM Symposium on Theory of Computing*, 1996. 6

[Gö31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme, i. *Monatshefte für Mathematik und Physik*, 1931. 2

[HA28] David Hilbert e Wilhelm Ackermann. *Principles of Mathematical Logic*. Springer-Verlag, 1º edição, 1928. 2

[Hil02] David Hilbert. Mathematical problems. international congress of mathematicians at paris in 1900. *Translated by Mary Winston Newson*, 1902. 2

[Kow06] Luis A. B. Kowada. *Construção de Algoritmos Reversíveis e Quânticos*. Tese de Doutorado, COPPE, Universidade Federal do de Campinas, Brasil, Março 2006. ix, 34

[KP07] Mozammel H.A. Khan e Marek A. Perkowski. Quantum ternary parallel adder/subtractor with partially-look-ahead carry. *Journal of Systems Architecture*, 2007. 37

[Lim05] Elon Lages Lima. *Espaços Métricos*. IMPA - Projeto Euclides, 4º edição, 2005. 10

[Meg08] Zdzislaw Meglicki. *Quantum Computing Without Magic*. Mit Press, 1º edição, 2008. ix, 5, 31

[MF11] Firouz Mosharraf e Behrouz A. Forouzan. *Fundamentos da Ciência da Computação*. Cengage Learning, 2º edição, 2011. 31

[Mon15] Ashley Montanaro. Quantum algorithms: an overview. *arXiv:1511.04206v2*, 2015. 40

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 1965. ix, 4

[NC00] Michael A. Nielsen e Isaac L. Chuang. *Quantum Computing and Quantum Information*. Cambridge University Press, 1º edição, 2000. ix, xi, 8, 9, 11, 12, 21, 32, 36, 43, 44

[NS16] Marcel Novaes e Nelson Studart. *Mecânica Quântica Básica*. Livraria da Física, 1º edição, 2016. 12

[OS04] Ivan S. Oliveira e Roberto S. Sarthour. Computação quântica e informação quântica. *V Escola do CBPF*, 2004. ix, xi, 5, 19, 20

[Pla] Google Play. Quantum circuit simulator. https://play.google.com/store/apps/details?id=mert.qcs&hl=pt_BR. último acesso em 30/09/2018. x, 44, 45

[PLeNM12] Renato Portugal, Carlile C. L. e Luiz M. Carvalho end Nelson Maculan. *Uma Introdução à Computação Quântica*. SBMAC, 2º edição, 2012. 17

[Qa] IBM Q. Ibm q expercience. https://quantumexperience.ng.bluemix.net/qx/editor. último acesso em 25/09/2018. x, 48, 49

[Qb] IBM Q. Ibm q quantum computing. http://www.research.ibm.com/ibm-q/. último acesso em 20/08/2018. 48

[RP10] Arushi Raghuvanshi e Marek Perkowski. Fuzzy quantum circuits to model emotional behaviors of humanoid robots. *IEEE Congress on Evolutionary Computation*, 2010. 41

[Sch09] Jon Schiller. *Quantum Computers*. Booksurge Publishing, 1º edição, 2009. 32

[Sel04] Peter Selinger. Towards a quantum programming language. *Math. Struct. in Comp. Science*, 2004. 52

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. Em *35th Annual Symposium on Foundations of Computer Science*, 1994. 6

[Sip05] Michael Sipser. *Introdução à Teoria da Computação*. Cengage Learning, 2º edição, 2005. 3

[Soa09] Mario Soares. *Cálculo de uma Variável Complexa*. IMPA, 1º edição, 2009. 55, 56

[Tur36] Allan Turing. On computable numbers with an application to the entscheidungsproblem. *C. F. Hodgson and Son*, 1936. 3

[VBE95] Vlatko Vedral, Adriano Barenco e Artur Ekert. Quantum networks for elementary arithmetic operations. *Physical Review*, 1995. ix, 33, 34

[VdRC06] Juliana Kaizer Vizzotto e Antonio Carlos da Rocha Costa. Towards quantum haskell via quantum arrows. *Workshop-Escola de Computação e Informação Quântica*, 2006. 52

[Öm02] Bernhard Ömer. Classical concepts in quantum programming. *arxiv.org/abs/quant-ph/0211100v2*, 2002. 51