Universidade Federal do Rio Grande do Sul

Instituto de Física

Estudo e Análise de Algoritmos Quânticos

Leandro Mengue da Silva

Trabalho de Conclusão de Curso realizado sob orientação da Professora Dra. Sandra Denise Prado

Porto Alegre
11/11/2013

Agradecimentos

Aos meus pais, em especial a minha mãe que em sua simplicidade sempre viu na educação e no estudo a melhor maneira dos filhos obterem um futuro melhor.

A todos professores do IF-UFRGS pelos ensinamentos, em especial aos professores Roberto da Silva, Silvio Dahmen, Acirete Simões e Sandra Prado.

Aos amigos Jorge Henrique, Vinicius Brei e Celso Matos pela amizade e exemplo acadêmico.

A minha esposa Silvia e meu filho Lucas pelo amor, carinho e apoio incondicional.

A Deus por colocar todas essas pessoas em minha vida.

Resumo

Em 1939 Linus Pauling[1] publicava o mais importante trabalho sobre química de todos os tempos e, utilizando a Mecânica Quântica, modificava o futuro da química fazendo-a avançar anos-luz. Da mesma maneira, nas décadas de 1980 e 1990, Richard Feynman[2], David Deutsch e Peter Shor apresentavam suas idéias e revolucionavam a área da computação com suas propostas sobre Computação Quântica. Nessas idéias, o algoritmo quântico tem papel fundamental, pois substitui a tradicional lógica booleana e os algoritmos clássicos por uma nova forma de pensar, baseada na Mecânica Quântica.

O objetivo deste trabalho é apresentar um estudo dos principais algoritmos quânticos, descrevendo seus componentes fundamentais como o *bit* quântico, as portas quânticas e os circuitos quânticos, de maneira a proporcionar elementos para efetuar uma análise do comportamento, performance e aplicação dos algoritmos.

Abstract

In 1939 Linus Pauling[1] published the most important work on chemistry every since and using Quantum Mechanics he has modified the chemical's future making it to move light years. Likewise, in the 1980s and 1990s, Richard Feynman[2], David Deutsch and Peter Shor presented their ideas and revolutionized the computing area with its proposals for Quantum Computing. In these ideas, quantum algorithms play a key role, because they replace the traditional boolean logic and classical algorithms for a new way of thinking, based on Quantum Mechanics.

The goal of this work is to present a study of the major quantum algorithms, describing its key components such as quantum bit, quantum gates and quantum circuits, in order to provide elements to perform an analysis of the behavior, performance and application of algorithms.

Índice

1. Introdução..... 5

2. Mecânica Quântica..... 6
2.1 Álgebra Linear e a Notação de Dirac..... 6
2.2 Postulados da Mecânica Quântica..... 7
2.3 Paralelismo e Emaranhamento..... 8

3. Computação Quântica..... 9
3.1 *Bit* Quântico..... 9
3.2 Portas Quânticas..... 10
3.3 Circuitos Quânticos..... 13
3.4 Algoritmos Quânticos..... 14

4. Algoritmos Baseados na TFQ..... 15
4.1 Transformada de Fourier Discreta 15
4.2 Transformada de Fourier Quântica..... 15
4.3 Estimativa de fase..... 16
4.4 Busca de ordem..... 16
4.5 Aplicação: Fatoração..... 19

5. Algoritmo de Grover..... 21
5.1 Busca Clássica em Lista de Elementos Desordenados..... 21
5.2 Busca Quântica..... 21

6. Algoritmo de Deutsch-Jozsa..... 23
6.1 Função Constante ou Balanceada..... 23
6.2 Verificação Quântica da Função Constante ou Balanceada..... 23

7. Análise dos Algoritmos..... 24
7.1 O Simulador qSimIF..... 24
7.2 Análise dos Algoritmos..... 25

8. Considerações Finais..... 27

9. Bibliografia..... 28

1. Introdução

A Mecânica Quântica é um ramo fundamental da Física com vasta aplicação, tendo seus alicerces estabelecidos na primeira metade do século XX por Planck, Bohr, de Broglie, Heisenberg, Schrödinger, Born, von Neumann, Dirac, Pauli, entre outros.

A Computação Quântica é uma proposta de aplicação prática da Mecânica Quântica, onde são propostos elementos como o computador quântico, portas quânticas, circuitos e algoritmos quânticos.

Segundo Nielsen e Chuang[3], um dos principais motivos do grande interesse na Computação Quântica são seus novos e espetaculares algoritmos, capazes de resolver problemas já solúveis, mas que classicamente requerem um quantidade exorbitante de recursos como tempo computacional.

Este Trabalho de Conclusão de Curso tem como objetivo estudar e analisar os mais importantes algoritmos quânticos, que são o algoritmo de Grover (utilizado em buscas), o algoritmo de Deutsch-Jozsa (verificação de funções constantes ou balanceadas) e os algoritmos baseados na Transformada de Fourier Quântica (busca de fase, busca de ordem e fatoração). Esses tópicos serão discutidos nos capítulos 4, 5 e 6.

Uma revisão dos princípios básicos da Mecânica Quântica, com foco naqueles mais fortemente ligados a Computação Quântica, será apresentada no capítulo 2.

A introdução do conceito de Computação Quântica é fundamental para o entendimento dos algoritmos quânticos, e isso será feito no capítulo 3.

A estrutura deste trabalho foi organizada de maneira que o leitor possa iniciar, evoluir e finalmente entender o complexo mundo do algoritmos quânticos. Para isso, focaremos em elucidar certos conceitos que muitas vezes não são tão bem explicados na literatura existente sobre o assunto e por vezes optaremos por desdobrar mais claramente um conceito em outros de mais fácil entendimento, utilizando uma proposta objetiva e evitando extensas formalizações ou deduções.

Nos dois últimos capítulos, apresentaremos a análise dos algoritmos e nossas considerações finais sobre este estudo dos algoritmos quânticos, bem como as possibilidades futuras a este trabalho.

2. Mecânica Quântica

A teoria quântica é conceitualmente muito rica e do mesmo modo tecnicamente difícil. Sua riqueza conceitual provém das idéias de como o mundo microscópico funciona e estas contrastam com o nosso pensamento clássico, pois este está baseado no mundo macroscópico em que vivemos. Sua dificuldade técnica por sua vez provém da necessidade de um formalismo matemático diferenciado e que, para muitos sistemas, não nos oferece soluções exatas.

A Mecânica Quântica é a teoria física de maior sucesso no estudo de sistemas físicos nos níveis atômico e sub-atômico. Ela oferece resultados excelentes para questões em que as teorias clássicas apresentam resultados ruins ou até mesmo absurdos como a radiação de corpo negro (Eisberg[4]), por exemplo.

Conhecer os conceitos básicos da Mecânica Quântica é fundamental para o entendimento da Computação Quântica. Alguns desses conceitos serão mostrados a seguir.

2.1 Álgebra Linear e a Notação de Dirac

A representação matricial tem fundamental importância na Mecânica Quântica, onde os estados quânticos são representados por vetores (matriz coluna) e os operadores (observáveis) são representados por matrizes quadradas.

Em Mecânica Quântica utiliza-se a Notação de Dirac (Cohen-Tannoudji[5]) para representar os estados (vetores) e estes são chamados *kets*:

$$|\psi\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \qquad |\phi\rangle = \begin{bmatrix} c \\ d \end{bmatrix}$$

Os observáveis (operadores) são matrizes como:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad B = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$$

É importante também a definição de produto tensorial entre dois vetores:

$$|\psi\rangle \otimes |\phi\rangle = \begin{bmatrix} a \\ b \end{bmatrix} \otimes \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ ad \\ bc \\ bd \end{bmatrix}$$

E de maneira semelhante para duas matrizes:

$$A \otimes B = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \otimes \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{bmatrix}$$

Os operadores utilizados serão baseados em matrizes unitárias, logo a definição da matriz $U^{\dagger}$ se torna fundamental neste contexto, onde $U^{\dagger}$ é a transposta conjugada de *U:*

$$U = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad => \quad U^{\dagger} = \begin{bmatrix} a^* & d^* & g^* \\ b^* & e^* & h^* \\ c^* & f^* & i^* \end{bmatrix}$$

Onde n^* é o conjugado de n, logo se $n=(x+yi)$ então $n^*=(x-yi)$.

2.2 Postulados da Mecânica Quântica

A Mecânica Quântica não é uma teoria física que pode ser compreendida pela referência a primeiros princípios, ou seja, a princípios básicos que são claros e distintos. Em vez disso, seus princípios foram postulados (Sakurai[6], Griffiths[7]) e aceitos até que se provem estar incorretos, pois essa teoria é a quem mais tem tido sucesso na física.

Postulado 1: Em um determinado instante de tempo, o sistema físico é descrito por um vetor de estado pertencente ao espaço de estados do sistema.

Postulado 2: Toda quantidade física mesurável é descrita por um operador que é um observável.

Postulado 3: O único resultado possível em uma medida de uma quantidade física é um dos autovalores do observável correspondente.

Postulado 4: A probabilidade de obter um determinado autovalor de um observável é igual ao módulo quadrado do autoestado associado ao autovalor.

Postulado 5: Se a medida de uma quantidade física de um sistema físico fornece um determinado valor, o sistema imediatamente após a medida é a projeção normalizada do vetor de estado.

Postulado 6: A evolução no tempo do vetor de estado de um sistema físico é governado pela equação de Schrödinger.

2.3 Paralelismo e Emaranhamento

Para um sistema de duas partículas, por exemplo, podemos ter um estado quântico resultante do produto tensorial entre dois outros estados:

$$|\varphi\rangle = |\psi\rangle \otimes |\phi\rangle$$

Porém, em situações mais gerais, os estados não podem ser descritos da forma acima, pois não são o resultado do produto tensorial entre dois outros estados. Um exemplo é:

$$|\Phi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

Este é o caso de um estado chamado de *estado emaranhado*, pois quando medimos o estado da partícula 1, digamos na posição 1 do *ket* $|00\rangle$, o estado da partícula 2 já fica automaticamente determinado. Logo os estados não são independentes. A existência dos estados emaranhados é a principal razão pela qual precisamos operar o estado quântico como um todo, sem poder dividi-lo em seus componentes para efetuar a operação separadamente. Isso faz com que a representação clássica do estado necessite um *ket* de dimensão 2^n. Deve ser observado que só podemos falar em emaranhamento para sistemas de duas ou mais partículas.

Uma consequência direta do emaranhamento é o *paralelismo quântico*, que acontece quando se realiza uma operação em uma superposição de estados emaranhados, ocasionando a aplicação da operação em todos os estados superpostos ao mesmo tempo. Este é um dos pontos chave utilizados pela computação quântica para obter um ganho exponencial de tempo nas operações.

3. Computação quântica

Desde 1936 quando Alan Turing[8] desenvolveu a noção abstrata de computador programável, a ciência da computação tem evoluído vertiginosamente a cada década com computadores cada vez mais potentes. Porém, em acordo com as previsões da Lei de Moore[9] de 1965, os computadores estão atingindo um limite devido a redução do tamanho de seus componentes, pois como estão cada vez mais miniaturizados, efeitos quânticos começam a interferir no funcionamento dos componentes.

Uma solução a este problema é mudar para um novo paradigma dado pela Computação Quântica, cuja base de funcionamento é a Mecânica Quântica. Essa relação é facilmente perceptível ao observarmos alguns dos componentes da Computação Quântica como o *bit* quântico e a porta quântica, pois sua definição segue exatamente os conceitos da Mecânica Quântica (onde são *ket* e operador).

3.1 *Bit* Quântico

A computação clássica está baseada no conceito fundamental do *bit*, que pode assumir o valor 0 ou 1. Assim, todos circuitos e portas clássicas estão baseadas neste funcionamento binário.

Assim como na computação clássica temos como conceito fundamental o *bit*, na Computação Quântica temos o *bit quântico*, também chamado de *q-bit* e que é representado por um vetor (*ket*) cujos estados puros são:

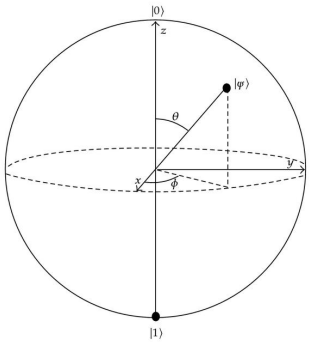
$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

O *q-bit* (assim como o *bit*) é um objeto matemático utilizado nas operações mais básicas da computação, porém a grande diferença entre o *bit* e o *q-bit* reside no fato de que o *bit* só pode assumir valores 0 ou 1, já o *q-bit* pode estar em estado de superposição de suas bases $|0\rangle$ e $|1\rangle$:

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

Outra característica importante do *q-bit* é que antes de uma medida ele pode estar em um estado de superposição (conforme acima), mas após a medida somente poderá resultar do estado $|0\rangle$ ou $|1\rangle$. Os coeficientes complexos α e β (na verdade seus módulos ao quadrado) correspondem a probabilidade de se obter o estado $|0\rangle$ ou $|1\rangle$ respectivamente, logo $P_{|0\rangle}=|\alpha|^2$ *e* $P_{|1\rangle}=|\beta|^2$ e ainda devemos ter sempre $|\alpha|^2 + |\beta|^2 = 1$.

Uma importante representação geométrica do *q-bit* pode ser feita através da esfera de Bloch, onde o ângulo θ está relacionado a proporção (em módulo ao quadrado) de α em relação a β e o ângulo Φ está relacionado ao sinal da parte imaginária de α e β.



Nesta representação, o vetor que representa o *q-bit* apontará mais para cima quando $|\alpha|^2 > |\beta|^2$ e mais para baixo quando $|\alpha|^2 < |\beta|^2$. Já o sinal da parte imaginária determinará se o vetor apontará mais para esquerda (em caso de sinal positivo) ou mais para a direita (em caso de sinal negativo).

A esfera de Bloch permite também avaliar a atuação dos operadores sobre o *q-bit*, podendo-se verificar a posição do vetor antes e depois da aplicação do operador e assim entender sua ação.

3.2 Portas Quânticas

Uma porta quântica é um operador que atua sobre um ou mais *q-bits*. Esta atuação ocorre de maneira matricial, onde os *q-bits* são representados por um vetor (*ket*) e a porta por um operador (matriz):

$$-\boxed{X}- \;=\; \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Se o operador atua sobre apenas um *q-bit*, ele é representado por uma matriz 2x2 unitária ($U^{\dagger}U=I$). Mas podemos ter portas quânticas que atuam em vários *q-bits*, logo esses operadores terão dimensão 2^n, onde n é o número de *q-bits*.

Um conjunto de portas é dito universal se qualquer computação puder ser expressa em termos dessas portas.

Um porta ou operação é dita controlada quando sua aplicação é condicionada a um determinado estado de um *q-bit*. Por exemplo, numa porta *U-Controlada* (), se o *q-bit* de controle (inferior) é $|1\rangle$, então é aplicada a operação *U* no *q-bit* alvo (superior).

A seguir estão listadas as mais importantes portas de um *q-bit* e sua aplicação.

 Porta Not ou Pauli-X

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow X(\alpha|0\rangle + \beta|1\rangle) = \beta|0\rangle + \alpha|1\rangle$$

 Porta Pauli-Z:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \Rightarrow Z(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle - \beta|1\rangle$$

 Porta Hadamard:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \Rightarrow H(\alpha|0\rangle + \beta|1\rangle) = \alpha\frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta\frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

 Porta Pauli-Y:

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

 Porta de Fase:

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

 Porta π/8:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \Rightarrow S(\alpha|0\rangle + \beta|1\rangle) = \alpha|0\rangle + e^{\frac{i\pi}{4}}\beta|1\rangle$$

 Porta R_k:

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

A seguir estão listadas as mais importantes portas de 2 e 3 *q-bits*:

 Porta Não-Controlado:

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

 Porta de Troca:

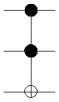
$$Troca = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

 Porta Z-Controlado:

$$CZ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

 Porta de Fase-Controlada:

$$CS = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

 Porta Toffoli:

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

 Porta Fredkin:

$$Fredkin = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Algumas das portas apresentadas tem um relevância tão grande que as descreveremos em mais detalhes, pois são utilizadas com muita frequência em circuitos quânticos.

 A porta *Not* tem a função de trocar os coeficientes α e β do q-bit. Em estados puros seu efeito é levar o estado |*0*> para |*1*> e o estado |*1*> para |*0*>, ou seja, inverter o *q-bit*.

 A porta *Hadamard* é utilizada para transformar um *q-bit* em uma superposição de estados da base |*0*> e |*1*>. Ela tem papel fundamental no uso do emaranhamento e paralelismo quântico, pois permite criar estados de superposição a partir de estados puros.

 A porta *Não-Controlado* tem a função de inverter (aplicar *Not*) o *q-bit* alvo (inferior) somente se o *q-bit* de controle (superior) estiver no estado |*1*>.

 A porta *Toffoli* é muito semelhante a porta *Não-Controlado*, porém utiliza dois *q-bits* de controle, ou seja, só inverte o *q-bit* alvo se os dois *q-bits* superiores estiverem no estado |*1*>.

3.3 Circuitos Quânticos

Um circuito quântico é um conjunto discreto de componentes (quânticos) que descrevem um processo computacional. Estes componentes são portas quânticas representadas por operadores, que quando dispostos de maneira apropriada permitem computar operações e implementar algoritmos.

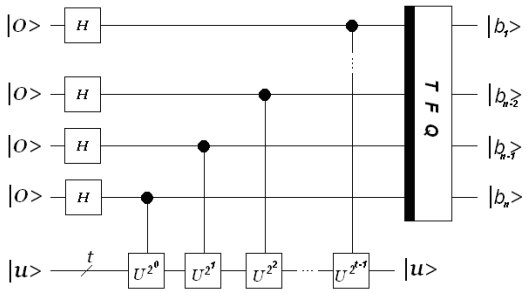
A representação gráfica de um circuito quântico nos permite avaliar seu funcionamento. Os principais elementos que o compõem são:

- *Q-bits* de entrada: expressam o estado inicial em que o sistema foi preparado.

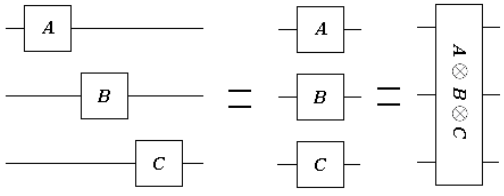
- Linhas de transporte: são os fios que transportam cada *q-bit*, conectando as postas quânticas. Um traço diagonal a linha indica que estão sendo representados vários fios. Pontos do tipo "..." representam uma expansão (oculta) do circuito.

- Portas quânticas: operadores que atuam em um ou mais *q-bits*. Uma parte do circuito (ou até mesmo todo ele) pode ser agrupado em um operador, pois a aplicação sucessiva de operadores tem como resultado também um operador (matriz) que quando aplicado efetua (de uma única vez) as mesmas operações que os operadores que o compõem. Se o operador for representado com uma barra na sua lateral esquerda, isso significa que atua na sua forma inversa: $U^{\dagger}$.

- *Q-bits* de saída: expressam estado final do sistema após as operações do circuito quântico.



Partes do circuito quântico podem ser agrupadas, formando um único operador com a mesma funcionalidade:

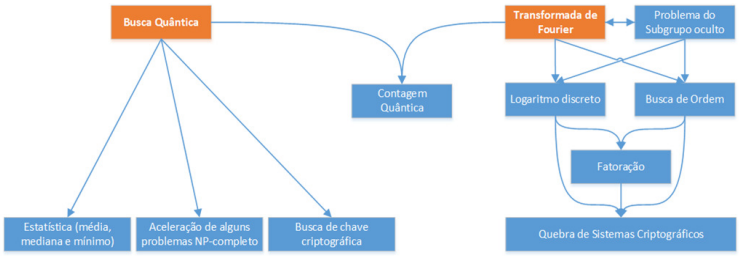


Na representação acima, caso um operador não esteja presente (por exemplo *B*), tendo apenas o fio em sua posição, ele deverá ser representado pela matriz identidade (ex.: $A \otimes I \otimes C$).

3.4 Algoritmos Quânticos

Os algoritmos quânticos distinguem-se dos algoritmos clássicos pelo fato de usarem propriedades quânticas não disponíveis nos computadores clássicos, como o paralelismo quântico e o emaranhamento.

Sua modelagem baseia-se na utilização de circuitos quânticos que por sua vez são constituídos de portas quânticas. Essa construção não é trivial, pois necessita uma mudança em nosso modo de elaborar algoritmos.



Ganho quadrático Ganho exponencial

Observando a figura acima, podemos nos questionar: porquê existem poucos algoritmos quânticos melhores que os clássicos? Segundo Nielsen e Chuang[3], isso é devido a três fatores:

1) Porque inventar bons algoritmos quânticos é, por natureza, um problema difícil.

2) Fazer bons algoritmos não é fácil. Um bom raciocínio lógico (e matemático) não é uma qualidade comum.

3) Porque pensamos classicamente e é necessário mudar nosso padrão de pensamento para lidar com algoritmos quânticos.

É interessante verificar que o limiar do que é um algoritmo quântico, um circuito quântico ou uma porta quântica só depende de como desejamos estruturar a sua organização e apresentação para uma maior clareza, isto porque um circuito, ou até mesmo todo algoritmo, pode ser representado por um único operador com dimensão resultante do produto tensorial de todas portas, ou seja, 2^n onde n é o número total de *q-bits*.

4. Algoritmos Baseados na TFQ

A Transformada de Fourier (TF) é uma ferramenta matemática muito utilizada em física, matemática, computação e engenharia. Suas utilidades englobam várias aplicações como remover ruídos de sequência de dados, examinar propriedades de cristais, produzir hologramas, etc. Ela é especialmente importante quando analisamos dados de fenômenos que tem alguma periodicidade subjacente, permitindo extrair o comportamento periódico oculto de uma função.

4.1 Transformada de Fourier Discreta

A Transformada de Fourier Discreta (TFD) é uma transformação matemática onde uma entrada com um vetor de N números complexos, x_0, ..., x_{N-1}, gera uma saída de outro vetor de N números complexos , y_0, ..., y_{N-1}, definido por:

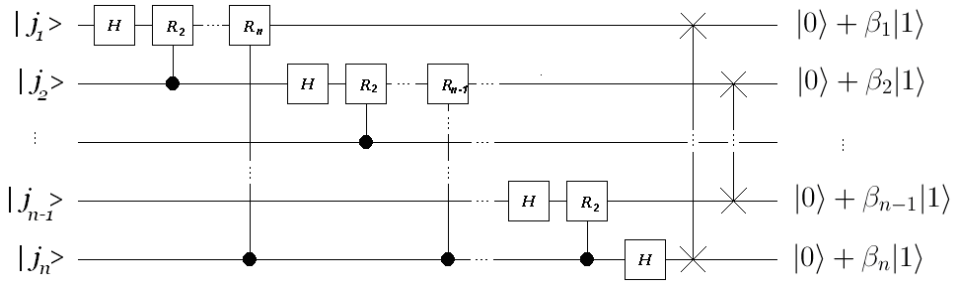
$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j . e^{2\pi i j k / N}$$

4.2 Transformada de Fourier Quântica

A Transformada de Fourier Quântica (TFQ) é uma transformação idêntica a TFD, porém utiliza elementos da mecânica quântica (expressos na notação de Dirac) para o cálculo do vetor (*ket*) de saída:

$$|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k / N} |k\rangle$$

Esta transformação poder ser implementada com o seguinte circuito quântico:



Onde $\beta_i = e^{2\pi i (0.j_i \ldots j_n)}$. Os *q-bits* de saída tem um fator de normalização $1/\sqrt{2}$ não mostrado no esquema do circuito, por uma questão de simplicidade.

O estado inicial pode ser definido como $|j> = |j_n 2^0, j_{n-1} 2^1, ..., j_1 2^{n-1}>$ com j_k podendo ser $|0>$ ou $|1>$, representando binariamente j. Assim, a cada q-bit é aplicada a porta Hadamard e uma sequência de portas R_k-Controladas. Ao final é feita uma operação de troca para ordenar os *q-bits*.

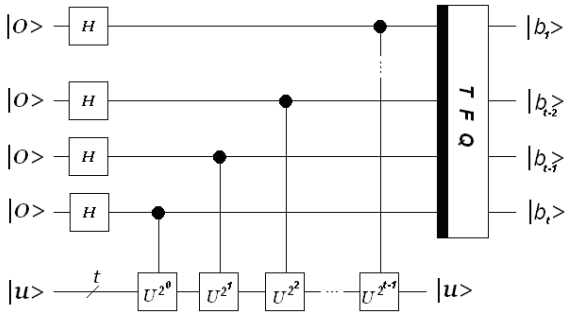
4.3 Estimativa de Fase

Consideremos um operador U com os seguintes autovetor e autovalor:

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle$$

Onde φ é desconhecido e representa uma fase a ser determinada.

A determinação da fase φ baseia-se na utilização da TFQ inversa e é a chave para outros algoritmos com a busca de ordem. Sua implementação é mostrada no circuito a seguir:



O circuito inicia criando uma superposição de estados através das portas Hamadard, logo após são aplicadas t operações U controladas. A última etapa é a aplicação da TFQ inversa, ficando o resultado da estimativa de fase φ medido nos primeiros t *q-bits* de saída.

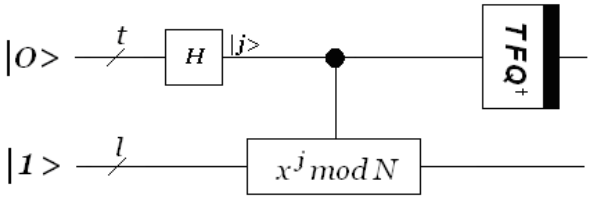
4.4 Busca de Ordem

A ordem de um determinado número (inteiro e positivo) x módulo N é o menor inteiro positivo r tal que:

$$x^r = 1 \,(mod\, N)$$

Onde *módulo N* ou *mod N* representa o *resto da divisão por N*.

A determinação de ordem é implementada no circuito abaixo:

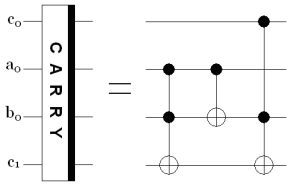


O circuito mostrado acima é uma visão geral do funcionamento do circuito quântico de determinação de ordem. Considerando a importância do algoritmo de busca quântica de ordem, a seguir explicaremos em detalhes a construção do algoritmo e dos circuitos utilizado nele, baseado nos trabalhos de Vedral[10], Van Meter[11] e Muthukrishnan[12].

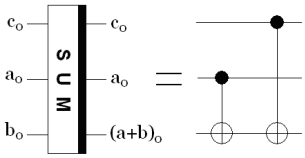
4.4.1 Operadores *Carry* e *Sum*

Uma das mais básicas operações largamente utilizadas em circuitos quânticos mais complexos é a soma. Para implementar a soma de estados representados por *q-bits*, necessitamos de dois operadores: *carry* e *sum*.

A porta *carry* tem como função avaliar três *q-bits* (a_0, b_0 e c_0) e identificar se o *q-bit* de alvo (c_1) deve ser colocado em $|1\rangle$ caso dois (ou três) dos três *q-bits* de entrada tenham valor $|1\rangle$.



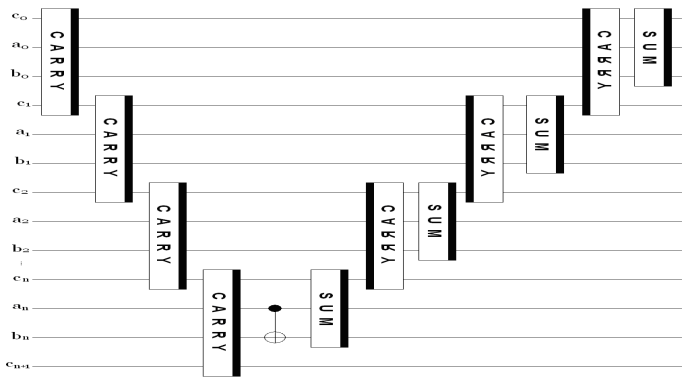
A porta *sum* tem como função avaliar três *q-bits* (a_0, b_0 *e* c_0) e colocar o resultado da soma módulo 2 em b_0.



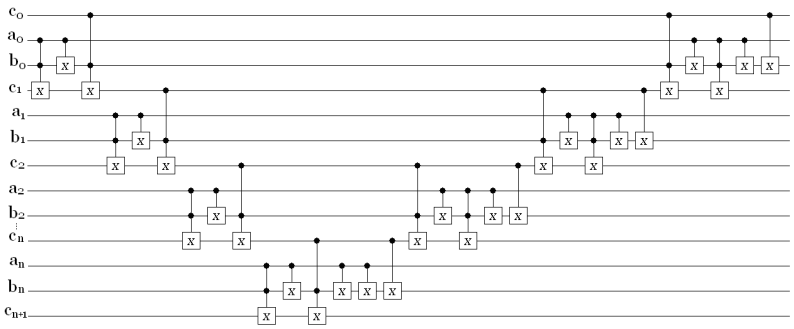
4.4.2 Circuito de Soma

O circuito de soma (*adder*) é utilizado para processar 2 conjuntos de *q-bits* e calcular a soma de *a+b*, colocando-a no conjunto de *q-bits* de saída *b*. Nele é necessário que tenhamos *n+1 carry q-bits* mais *2n q-bits* para representar *a* e *b*.

Este circuito é uma sequência de portas *carry* seguidas de uma sequência de portas *sum*, conforme mostrado abaixo:

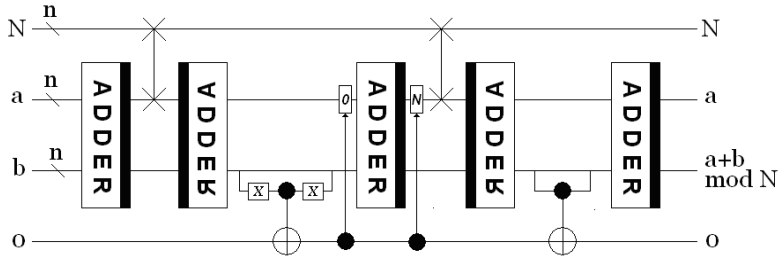


Cada porta do circuito de soma pode ser expandido em termos das portas básicas de seus componentes:



4.4.3 Circuito de Soma Módulo *N*

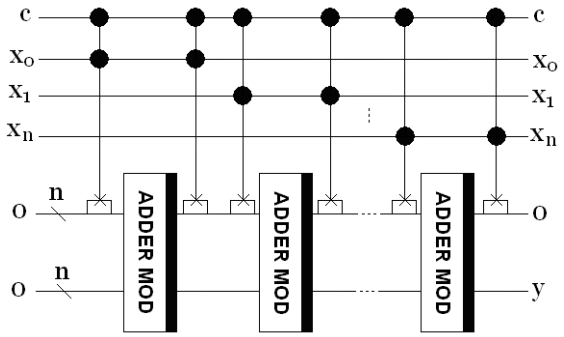
Com o circuito de soma podemos montar outro importante circuito quântico: *a+b módulo N*, que efetua a soma de *a+b* e obtém o resto da divisão por *N* e coloca no registro *b*. Assim teremos, por exemplo, *5+2 mod 3 = 1*.



O circuito de soma módulo *N* (onde *a* e *b* < *N*), tem basicamente duas partes distintas: a primeira verifica se *a+b* é maior que *N* (verifica se há *overflow*) e guarda a informação no *q-bit* auxiliar e a segunda excuta a soma *a+b*, subtraindo *N* se necessário. As portas Não-Controladas entre o segundo e o terceiro *adder* e entre o quarto e o quinto *adder* trabalham apenas com o *q-bit* mais significativo que conterá a informação do *overflow*.

4.4.4 Circuito de Multiplicação Módulo *N* Controlado

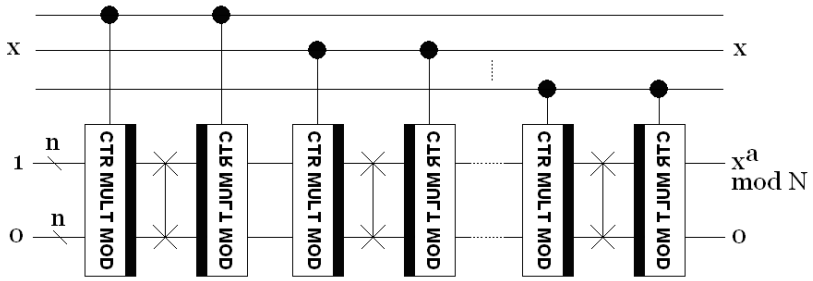
Agrupando-se *n* portas de soma módulo *N*, podemos construir o circuito de multiplicação módulo *N* controlado, que efetuará a operação *y=ax mod N*.



Neste circuito precisaremos de *n* repetições controladas da porta de soma módulo *N*, onde *n* é o número de *q-bits* suficientes para representar binariamente *ax* . O resultado estará representado nos *q-bits y*.

4.4.5 Circuito de Exponenciação Módulo *N*

Finalmente a construção do circuito quântico para calcular x^a módulo *N* é possível através do agrupamento de *2n* portas de multiplicação módulo *N*.



Cada uma das portas de multiplicação controlada executa uma operação de multiplicação por a^{2j} e a^{-2j} sucessivamente.

O circuito de exponenciação módulo *N* é o principal operador do algoritmo de busca de ordem, sendo sua construção nada trivial e pouco abordada na literatura existente sobre o assunto. Outros detalhes de sua modelagem podem ser encontrados em Vedral[10].

4.5 Aplicação: Fatoração

A fatoração de um número inteiro positivo *N* em seus divisores primos é um importante problema da matemática, como explicitado no trabalho de Gauss[13] de 1801:

> *"O problema de se distinguir números primos de números compostos, e a decomposição de números compostos em fatores primos, é um dos mais importantes e úteis de toda aritmética. (...) A grandeza da ciência parece exigir que seja cultivada a busca para a solução desse célebre problema".*

Sua resolução para números muito grandes requer muitos recursos computacionais em um computador clássico e o torna praticamente inviável de resolvê-lo. Essa dificuldade é o ponto chave de importantes sistemas criptográficos como o RSA[14].

O algoritmo quântico de fatoração de Shor[15] é, na verdade, um algoritmo clássico com uma etapa baseada no algoritmo quântico de busca de ordem. Seus passos estão descritos a seguir.

Entrada: *N* (número inteiro e composto pela multiplicação de dois números primos desconhecidos)

Saída: Um dos fatores de *N*

Passos:

1) *Se N é par, resultado = 2.*
2) *Se* $N = a^b$ *(com* $a>=1$ *e* $b>=2$*), resultado = a.*
3) *Escolher um número x aleatório no intervalo 1 a N-1, se* $mdc(x,N)\neq 1$ *resultado =* $mdc(x,N)$*.*
4) *Executar a rotina quântica de busca de ordem (*$x^r \bmod N=1$*).*
5) *Se r é par e* $x^{r/2}\neq -1 \pmod N$*, calcular* $mdc(x^{r/2}-1,N)$ *e* $mdc(x^{r/2}+1,N)$ *e verificar se um deles é fator de N. Caso contrário execute o algoritmo novamente a partir do passo 3.*

5. Algoritmo de Grover

5.1 Busca Clássica em Lista de Elementos Desordenados

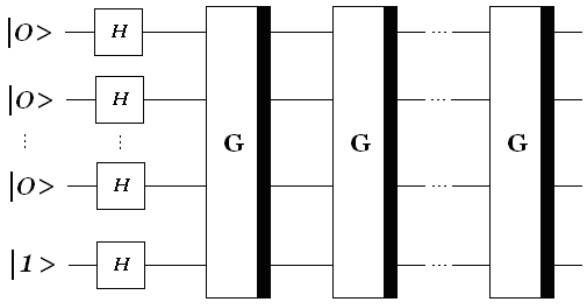
Em um computador clássico, a busca de um determinado elemento de uma lista de *N* elementos desordenada é feita pela procura sequencial de *1* até *N* ou até mesmo por uma busca aleatória. Em ambos os casos, o número médio de buscas necessárias é *(N+1)/(M+1),* onde *M* é a quantidade de elementos da lista iguais ao elemento procurado. Para uma lista com elementos que não se repetem, essa média é *(N+1)/(1+1) = (N+1)/2*.

É muito comum encontrar na literatura que a quantidade de buscas média necessária para achar um elemento de uma lista com 4 elementos é *2,25*. Se efetuarmos o cálculo acima, verificaremos que o número médio é diferente: *2,5 = (4+1)/(1+1) = 5/2*. Essa diferença se deve ao fato da diferença entre *"saber onde está"* e *"encontrar"* o elemento, pois no primeiro caso chegamos a *2,25* devido ao fato que, por exemplo, na lista *{1,2,3,4}* se desejamos encontrar o número 4, precisaremos apenas 3 buscas, pois após verificar *N-1* elementos e não achar, não precisamos olhar o último elemento para saber que lá está o procurado, assim após 3 buscas sabemos *"onde está"* o elemento 4 procurado. No segundo caso, consideramos que para *"encontrar"* o elemento 4 precisaremos fazer 4 buscas, pois na 4ª busca olharemos e encontraremos o elemento desejado.

5.2 Busca Quântica

O algoritmo de Grover[16] possibilita encontrar um elemento em uma lista desordenada com maior rapidez do que a busca clássica. Ele é baseado na aplicação de um oráculo (um operador) responsável por identificar a resposta correta.

O oráculo é baseado em uma função *f(x)* que retorna 1 se *x* corresponde ao valor procurado e 0 caso contrário.

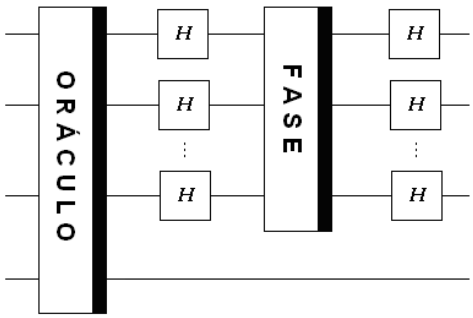


O algoritmo inicia colocando *n q-bits* no estado $|0\rangle$ e aplicando-lhes a porta Hadamard, o que coloca todos *q-bits* no estado de superposição $(|0\rangle + |1\rangle)/\sqrt{2}$.

Também é inicializado um q-*bit* auxiliar no estado |*1*> e aplicado a porta Hadamard, ficando no estado de superposição *(|0*> - |*1*>*)/*$\sqrt{2}$.

Após é aplicado o operador de Grover repetidas vezes, até que a probabilidade de se obter a resposta correta seja 1. No trabalho de Portugal[17] há uma bela (e extensa) demonstração de que o número de repetições necessárias é $\sqrt{N}$.

O operador de Grover é definido pelo circuito abaixo:



O oráculo é um operador que executa a transformação |*x*>|*q*> => |*x*>|*q*+*f(x)*>, ou simplificadamente |*x*> => *(-1)*$^{f(x)}$|*x*>.

A *fase* é um operador que executa a transformação |*x*> => -|*x*> para todos casos que |*x*> for diferente de |*0*>.

O resultado do algoritmo de Grover aparecerá nos *n* primeiros *q-bits* de saída do circuito, após a aplicação das sucessivas operações *G*. Estes *q-bits* representarão, através de estados puros |*0*> ou |*1*> , o valor procurado.

6. Algoritmo de Deutsch-Jozsa

6.1 Função Constante ou Balanceada

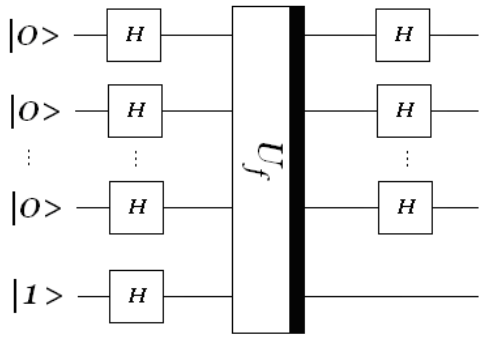
Considere um função *f(x)*, com $x \in (0,\ldots,N=2^{n+1})$ que pode ser definida com um dos comportamentos a seguir: ou ela é *constante* e sempre retorna valor 1 ou ela é *balanceada* e para metade dos valores de *x* retorna 0 e para outra metade retorna 1.

Um algoritmo clássico precisa de *2* a *N/2+1* execuções para determinar com certeza se a função é constante ou balanceada, pois necessita obter dois valores distintos (e concluir que a função é balanceada) ou examinar mais da metade dos resultados de *f(x)* e concluir se a função é constante.

6.2 Verificação Quântica da Função Constante ou Balanceada

O algoritmo de Deutsh-Jozsa[18] utiliza o paralelismo quântico e a interferência para executar a verificação da função *f(x)* e determinar se ela é constante ou balanceada com apenas uma avaliação de *f(x)*.

O circuito quântico que implementa o algoritmo de Deutsh-Jozsa é mostrado abaixo, onde U_f é uma transformação unitária utilizada para calcular *f(x)*.



O algoritmo inicia colocando *N q-bits* no estado |0> e aplicando-lhes a porta Hadamard, o que coloca todos *q-bits* no estado de superposição $(|0\rangle + |1\rangle)/\sqrt{2}$. Também é inicializado um *q-bit* auxiliar no estado |1> e aplicado a porta Hadamard, ficando no estado de superposição $(|0\rangle - |1\rangle)/\sqrt{2}$.

Após é aplicado o operador U_f que executará a transformação $|a\rangle|b\rangle \Rightarrow |a\rangle|b \oplus f(x)\rangle$, onde ⊕ representa adição módulo 2. Na sequência é aplicada a porta Hadamard novamente aos *N q-bits*, o que deixará cada um no estado puro |0> ou |1>.

O resultado da verificação é obtido através da medição dos *N q-bits* finais, onde a função *f(x)* será constante caso todos estados sejam iguais a |0>. Caso algum estado seja igual a |1> a função será balanceada.

7. Análise dos algoritmos

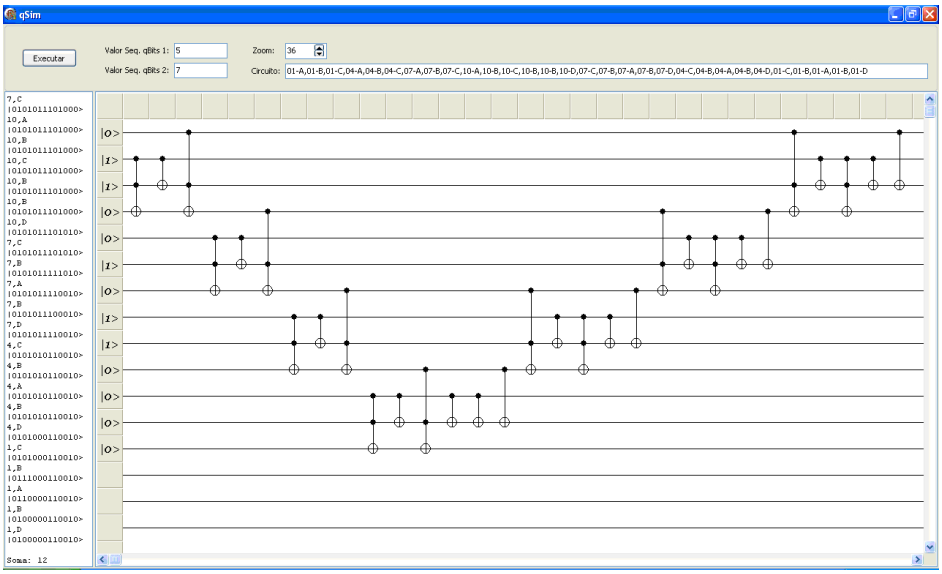
Analisar os algoritmos quânticos implica em entender com razoável profundidade os circuitos quânticos que os compõem. Para que isso fosse possível, decidimos fazê-lo com o auxílio de um simulador de circuitos quânticos, pois assim poderíamos verificar e acompanhar seu funcionamento passo a passo, bem como montar exemplos diversos dos circuitos.

7.1 O Simulador *qSimIF*

Em Barbosa[19] encontramos uma análise de vários simuladores de circuitos quânticos, onde verificamos que o simulador *Zeno* é apontado como um dos melhores simuladores, assim ele foi escolhido para auxiliar no inicio das análises. Porém, quando testamos um circuito quântico com mais de 10 *q-bits*, o simulador travava. Isso certamente é devido ao consumo de recursos de um circuito de 10 *q-bits* ser muito alto, pois nesse caso as portas quânticas são operadores representados por matrizes de dimensão $2^{10} \times 2^{10} = 2^{20} > 1$ *milhão* de elementos, resultado em 2^{30} *(1 bilhão)* de multiplicações quando aplicamos uma porta a um *ket* de *10 q-bits*.

Devido a esse problema de desempenho do simulador *Zeno*, decidimos desenvolver um protótipo próprio de um simulador de circuitos quânticos: o *qSimIF*.

No *qSimIF* conseguimos, por exemplo, construir e simular um circuito quântico (*Adder* com 13 *q-bits*) que soma dois conjuntos de *4 q-bits* que representam dois números inteiros.



A decisão de iniciar o desenvolvimento do simulador ocorreu devido a três fatores:

1) Os simuladores já existentes (que foram testados), não funcionavam satisfatoriamente para circuitos quânticos com mais de *10 q-bits*.
2) Auxiliaria na análise e testes dos algoritmos, facilitando a construção e visualização dos circuitos.
3) Para construir o simulador quântico estaríamos obrigados a entender em detalhe o funcionamento das portas quânticas, pois teríamos que construí-las!

No protótipo do simulador de circuitos quânticos foram implementadas algumas das funcionalidades básicas para a operação e testes tais como:
- Processamento dos comandos de montagem do circuito para as principais portas quânticas.
- Execução passo a passo do circuito (aplicação dos operadores).
- Saída gráfica do circuito na escala desejada (zoom).

7.2 Análise dos Algoritmos

A análise do algoritmo de fatoração de Shor é pertinente, pois nele está incluso o algoritmo de busca de ordem que por sua vez é baseado na estimativa de fase. Assim, comparando sua performance com o melhor algoritmo de fatoração clássico, identificamos um ganho exponencial, pois o algoritmo de Shor requer $(\log 2^n)^3$ operações contra $\exp(c(n^{1/3}(\log n)^{2/3})$ no caso clássico, onde c é uma constante e $2^n=N$. Isso significa que um computador quântico pode fatorar um número muito grande em alguns minutos, enquanto um computador clássico demoraria algumas dezenas de milhares de anos.

A implementação do algoritmo de Shor apresentada neste trabalho, utiliza um circuito quântico com $7n+1$ *q-bits*, ou seja, para fatorar um código RSA de *512 bits* seria necessário um computador quântico capaz de suportar um circuito quântico de *3.585 q-bits*. Existem variantes desta implementação do algoritmo de Shor que o fazem com $2n+3$ *q-bits* (Beauregard[20]), necessitando nesse caso um circuito com apenas *1.027 q-bits*, porém este circuito envolve vários operadores TFQ o que o deixa bem mais complicado, sendo um exemplo pouco didático.

É interessante comparar a quantidade de *q-bits* necessária para a fatoração de um exemplo real de um código RSA de *512 bits* (necessita *1027 q-bits*) com a capacidade do mais potente computador quântico atualmente disponível: o *Reinier* da empresa canadense *DWave*, que pode processar um registro de *512 q-bits*. Ou seja, ainda não é possível utilizá-lo para esta tarefa, mas a previsão é que em 1 ou 2 anos esses computadores ultrapassem a casa dos *Kq-bits* de processamento.

O algoritmo de busca de Grover, com seu funcionamento baseado na avaliação pelo oráculo, pode resolver problemas com ganho quadrático em

relação às buscas clássicas, necessitando $\sqrt{N}$ avaliações contra *N/2* no caso clássico. Esta performance não permite resolver problemas teóricos muitíssimo grandes, como uma busca em uma lista de 10^{100} elementos, pois necessitaria ainda 10^{50} avaliações, o que é inviável para qualquer computador quântico ou clássico. Porém, para casos reais onde temos muitos elementos e que classicamente podemos resolver com muito esforço computacional como por exemplo uma busca em uma lista de 10^{10} elementos, o algoritmo de Grover reduz o esforço computacional para um patamar (10^{5} avaliações) onde o problema pode ser resolvido com mais facilidade, em um tempo de execução aceitável.

O algoritmo de Deutch-Jozsa não tem análogo clássico, mas seu funcionamento demonstra perfeitamente o poder da utilização do paralelismo quântico, resolvendo o problema da função constante ou balanceada com apenas uma avaliação, enquanto classicamente necessitaríamos *(N/2)+1* verificações.

8. Considerações Finais

Os algoritmos quânticos são uma poderosa ferramenta computacional a serem utilizados em computadores quânticos, permitindo que estes realizem tarefas muito mais rápidas que os computadores clássicos.

Um grande impacto acontecerá na área da criptografia e segurança quando os computadores quânticos operarem com alguns milhares de *q-bits*, pois o algoritmo de fatoração de Shor poderá ser executado para números muitos grandes e proporcionará a quebra de chaves de segurança em poucos minutos.

Da análise do algoritmo de busca de Grover, identificamos a possibilidade de sua aplicação em algoritmos genéticos para incrementar a velocidade da etapa de seleção, pois esta etapa é a que mais consome recursos computacionais, que poderiam ser reduzidos quadraticamente: em uma população de 700 indivíduos seriam necessárias apenas 27 buscas para determinar o melhor, significando um tempo total 26 vezes menor. O desenvolvimento do oráculo para identificar as melhores soluções é o ponto chave desta aplicação.

Também verificamos que para a compreensão em detalhes dos algoritmos quânticos, uma ferramenta de simulação de circuitos quânticos é ferramenta fundamental, pois permite acompanhar cada ação de cada porta quântica (operador) sobre os *q-bits*. Além disso, a montagem do circuito quântico é facilitada com a utilização do simulador, permitindo uma perfeita visualização e navegação pelo circuito.

Dada a importância do simulador no entendimento dos algoritmos quânticos, o desenvolvimento de uma versão completa com mais recursos de interface e processamento é fundamental para trabalhos posteriores que visem analisar (e até mesmo criar) algoritmos quânticos.

9. Bibliografia

[1] L. C. Pauling. *The Nature of the Chemical Bond and the Structure of Molecules and Crystals*. Oxford University Press, London, 1939.

[2] R. P. Feynman. *Simulating physics with computers*. International Journal of Theoretical Physics, 21(6&7):467–488, 1982.

[3] M. A. Nielsen e I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 2005.

[4] R. M. Eisberg. *Física Quântica*. Editora Campus, Rio de Janeiro, 1983.

[5] C. Cohen-Tannoudji. *Quantum Mechanics*. Wiley, New York, 1977.

[6] J. J. Sakurai. *Modern Quantum Mechanics*. Addison-Wesley Pub. Co., Reading, 1994.

[7] D. J. Griffiths. *Introduction to Quantum Mechanics*. Prentice Hall, Englewood Cliffs, 1995.

[8] A. Turing. *On Computable Numbers with an Application to the Entscheidungsproblem*. C. F. Hodgson & Son, London, 1936.

[9] G. E. Moore. *Cramming More Components onto Integrated Circuits*. Electronics Magazine (April 19), p. 114–117, 1965.

[10] V. Vedral. *Quantum Networks for Elementary Arithmetic Operations*. http://arXiv.org/quant-ph/9511018, 1995.

[11] R. D. Van Meter. *Architecture of a Quantum Multicomputer Optimized for Shor's Factoring Algorithm*. http://arXiv.org/quant-ph/0607065, 2006.

[12] A. Muthukrishnan. *Classical and Quantum Logic Gates: An Introduction to Quantum Computing*. Quantum Information Seminar, Rochester Center for Quantum Information, 1999.

[13] C. F. Gauss. *Disquisitiones Arithmeticæ*. Fleischer, Lepzig, 1801.

[14] R. Rivest e A. Shamir e L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM 21, p. 120-126, 1977.

[15] P. W. Shor. *Algorithms for quantum computation: Discrete logarithms and factoring*. Proceedings 35th Annual Symposium on Foundations of Computer Science, p. 124–134, 1994.

[16] L. K. Grover. *A Fast Quantum Mechanical Algorithm for Database Search*. Twenty-Eighth Annual ACM Symposium on Theory of Computing, p. 212–219, 1996.

[17] R. Portugal. *Uma Introdução à Computação Quântica*. SBMAC, São Paulo, 2004

[18] D. Deutsch e R. Jozsa. *Rapid Solutions of Problems by Quantum Computation*. Proceedings of the Royal Society 439, p. 553–558, 1992.

[19] A. A. Barbosa. Um Simulador Simbólico de Circuitos Quânticos. Universidade Federal de Campina Grande, 2007.

[20] S. Beauregard. *Circuit for Shor´s algorithm using 2n+3 qubits*. Quantum Information and Computation, 3:175–185, 2003.