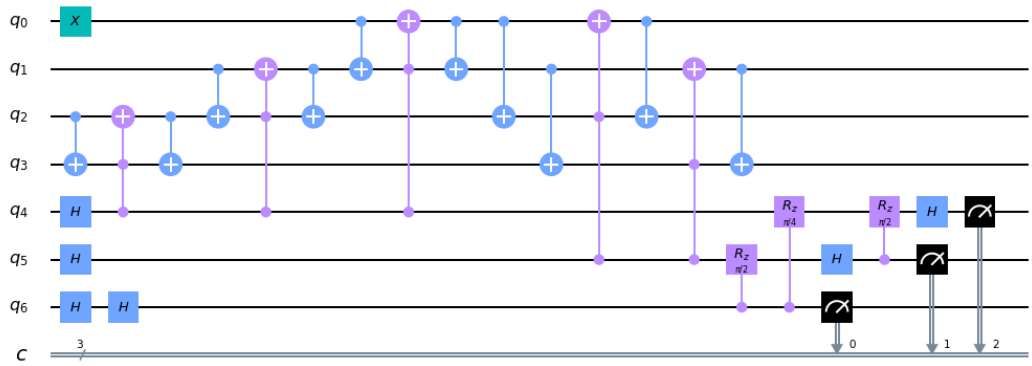
XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

Projeto Principal: “Algoritmo quântico de busca de Grover e de fatoração de Shor: implementação da simulação em computadores clássicos e quânticos”
Grande Área de conhecimento: Ciências Exatas e da Terra
Área de Conhecimento: Física
Subárea de Conhecimento: Física da Matéria Condensada
Aluno(a): Daniel Benvenutti (169448) - `danielgb23@gmail.com`
Orientador: Francisco Rouxinol - `rouxinol@ifi.unicamp.br`
Instituição: Instituto de Física Gleb Wataghin, UNICAMP
Palavras chaves: Eletrodinâmica Quântica, Computação Quântica e Informação
Referente: Relatório XXVIII Congresso de Iniciação Científica da UNICAMP

Resumo

Resumo do projeto original: Nós apresentamos um projeto de pesquisa focado no desenvolvimento pelo aluno de um conjunto de ferramentas para simular algoritmos quânticos em computadores clássicos e quânticos. Serão tratados e discutidos os conceitos básicos de computação quântica e mecânica quântica, como também a implementação de portas quânticas e o algoritmo de busca de Groover e de Shor. Com a implementação deste projeto é esperado que importantes conceitos de quântica, como medida, funções de onda de muitos corpos, e momento angular, sejam aprofundados como também o estudo de importantes tópicos avançados em física e engenharia.

Resumo Ilustrado



Computação quântica combina ciência da computação com mecânica quântica, possibilitando a realização de certos cálculos exponencialmente mais rápido do que as contrapartes clássicas, fazendo uso da superposição quântica e emaranhamento de estados quânticos Neste circuito apresentamos o Algoritmo de Shor para fatorar C=15 com a=2 que foi executado no computador quântico da IBM *ibmq 16 melbourne*. Neles temos 3 Hadamards de superposição nos qubits do expoente, um circuito de CNOTs e Toffollis para multiplicar a base e o circuito de transformada de Fourier quântica inversa nos qubits do expoente também

1 Atividades Principais

Inicialmente desenvolvemos os registradores de N-qubits. Cada registrador guardava a informação do estado composto por 3 sistemas de dois níveis (qubits), descrito por um único estado quântico conjunto $|\psi\rangle$. Utilizando estes vetores, preparamos algorítimos para preparar o sistema de interesse no estado de interesse.

Na segunda etapa, desenvolvemos funções que simulam uma porta quântica (análogo a uma porta lógica), necessárias para operar os qubits. Utilizando os elementos dos vetores de estado e a portas lógicas, a probabilidade de medir um determinado valor pode ser determinado, simulando-se uma medida no sistema quântico.

Com estas funções e algorítimos programados, foram preparados os algorítimos de Grover de busca e Shor de fatoração de números inteiros. Ambos funcionaram como esperado.

Na última implementamos os algorítimos Groover de busca e Shor de fatoração de números inteiros no computador quântico da IBM e comparamos com os resultados obtidos em nosso programa.

Fornecemos o código utilizando nesta etapa na plataforma GitHub no endereço: `https://github.com/Danielgb23/ic_comp_quantica`

1.1 Registrador Quântico

A primeira parte do projeto e construir o simulador. Para simular o estado de N qubits precisamos de um vetor de 2^N elementos. Cada estado nesse vetor representa uma combinação das medidas possíveis dos qubits e superposições das mesmas. A probabilidade de cada medida é representada pelo quadrado do módulo(complexo) de cada elemento do vetor.

1.2 Portas quânticas

Uma das portas quânticas mais importantes é a *Porta de Hadamard*, H. Ela é extremamente interessante, pois coloca um qubit em uma superposição de dois estados.

Outra importante porta é a *Porta de Fase*, S. Ela muda a fase complexa do elemento $|1\rangle$ do vetor que descreve o qubit.

Portões CNOT são portas quânticas NOT (ou NÃO) controladas e funcionam da seguinte maneira: Há um qubit controlador e um qubit que sofre a operação NOT. Se e somente se o qubit controlador é $|1\rangle$ que o outro qubit é invertido . Se o controlador for $|0\rangle$ o controlado mantém o seu estado. O qubit controlador não é alterado. Com essa porta podemos fazer o estado de emaranhamento quântico.

1.3 Algoritmo de Groover

O algoritmo de Groover é um algoritmo de busca. Dado um vetor de possíveis respostas, procuramos neste vetor, um valor específico, que indica qual é a resposta correta. Com uma algoritmo normal de varredura em um computador clássico no pior caso temos que verificar todos os n elementos. Já no algoritmo de Groover precisamos de apenas $\sqrt{n}$ buscas. O algoritmo de Groover funciona usando uma técnica chamada amplificação de amplitude. Que tem esse nome pois o algoritmo amplifica a amplitude da resposta no vetor de estados.

O construção dele é baseado em dois estágios. Primeiro colocamos todos os qubits em superposição (todas as medidas com mesma probabilidade). Depois

vem o oráculo e o bloco de difusão de Grover. Que temos que repetir juntos $\frac{\pi}{4}2^{N_{qubits}}$ vezes (arredondado). O oráculo inverte a amplitude quântica da resposta e o bloco de difusão amplifica o que estiver invertido.

Em seguida avaliamos o desempenho da simulação do algoritmo de Groover no meu computador. Abaixo os resultados:

Qubits	tempo
1	1.20s
2	1.15s
3	1.36s
4	3.51s
5	29.33s
6	353.69s

Tabela 1: Tempos de simulação de Grover para diferentes números de qubits

Depois o algoritmo de Groover foi executado utilizando-se matrizes esparsas. Que são muito úteis nessas simulações, já que as matrizes dos operadores têm muitos zeros.

Qubits	tempo
1	0.86s
2	0.92s
3	0.91s
4	0.97s
5	1.04s
6	1.16s

Tabela 2: Tempos de simulação de Grover para diferentes números de qubits utilizando matrizes esparsas

Nota-se a grande melhoria no desempenho e como é possível utilizar um número muito maior de qubits com o mesmo computador.

1.4 Algoritmo de Shor

O algoritmo de Shor é um algoritmo de fatoração de números inteiros. Um dos seus passos é acelerado se executado em um computador quântico. O resto dos passos são feitos rapidamente em um computador clássico. A algoritimo se fundamenta no fato de que dado um C que se deseja encontrar os fatores e um a escolhido segundo algumas regras se encontrarmos um período p da função $a^p \, mod \, C$ par e não satisfaça $a^{p/2} \equiv -1 \, mod \, C$. $P_{\pm} = mdc(a^{p/2} \pm 1, C)$ são fatores não triviais de C. A parte quântica encontra p.

Para essa parte quântica dividimos os qubits em expoente e base. Depois executamos três divisões do circuito: a primeira faz a superposição de todos os qubits do expoente(para entrar como todos os valores possíveis ao mesmo tempo).

Depois multiplicamos a base que está inicializada em $|1\rangle$ sucessivamente par cada qubit do expoente por $a^{2^W V_{qubit}}$ onde 2^W e o peso binário do qubit (exemplo $101_b = 2^2 + 1 = 5_d$ então o peso do primeiro qubit é 4) e V_{qubit} é zero ou um dependendo do valor desse qubit.

Finalmente aplicamos a transformada quântica de Fourier inversa(IQFT). O que obtemos das operações anteriores é uma função semelhante a um pente em uma

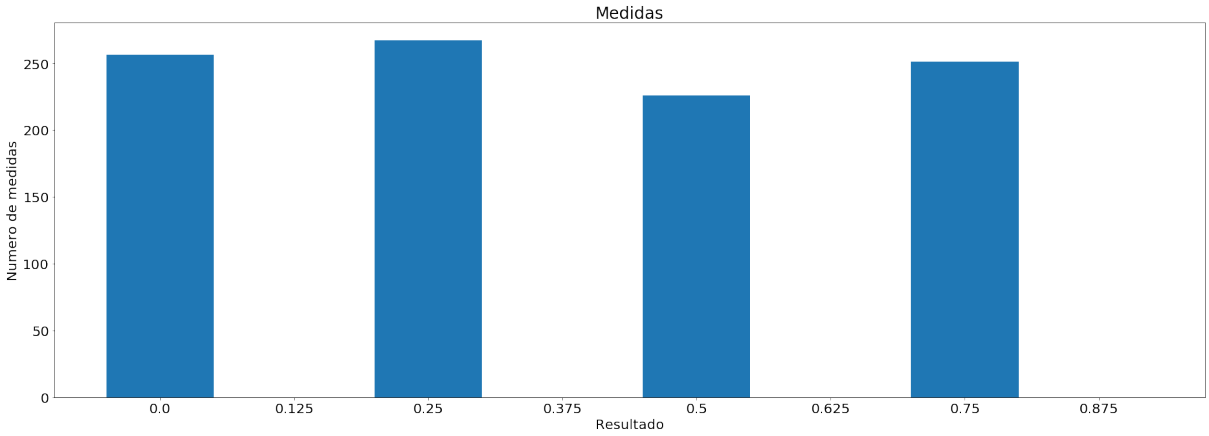


Figura 1: Histograma com valores obtidos para a aplicação do algoritmo de Shor com 7 qubits fatorando 15 com a=2

superposição dos vários valores para a base e com um período ω. A IQFT muda-a para o domínio da frequência e obtemos a frequência equivalente ao inverso do período $f = 1/\omega$ e suas harmônicas (2f, 3f, ...) como resultado(ver figura 1 com período 4 e frequência de 1/4). [1]

1.5 Implementação no computador Quântico

Após a simulação do qubits foram utilizados os computadores quânticos disponibilizados pela IBM na IBM Quantum experience [2]. Foram demonstrados os Algoritmos de Groover e Shor executados nessas máquinas. Tiveram de ser utilizadas novas portas quânticas enquanto outras não estão disponíveis na mesma forma da simulação no computador apresentando desafios. Outro desafio é a limitação física das máquinas da IBM. A linguagem utilizada para programar essas máquinas foi o Qiskit que é baseada em python.

1.5.1 Algoritmo de Groover

Para executar o algoritmo de Groover na máquina da IBM temos duas limitações principais. A primeira é que o algoritmo fica longo a medida que aumentamos o número de qubits. Isso porque precisamos repetir uma parte do circuito $\pi/4\sqrt{2^N}$ vezes arredondado onde N é o número de qubits. O que já deixa o circuito longo para a máquina da IBM se usarmos 3 qubits. E dependendo da máquina que utilizarmos como a IBM Q 14 Melbourne que tem mais qubits, portanto é mais suscetível a ruídos, já será muito mais difícil de distinguir a solução.

A segunda limitação são as portas que podemos utilizar. Numa simulação com matrizes fica simples fazer uma porta que inverte a fase de apenas um elemento da base mas agora temos que implementar isso com outras portas quânticas fundamentais. Felizmente para três qubits podemos usar uma porta Z controlada por dois qubits que pode ser feita com uma porta de Toffoli e duas de Hardamard nas entradas do qubit que será negado. Essa porta inverte a fase do componente $|111\rangle$ apenas. Para usar em outros componentes podemos colocar portas NOT na entrada e saída dessa porta para que mude a fase de outra combinação de qubits.

1.5.2 Algoritmo de Shor

No algoritmo de Shor a maior limitação para a implementação em um computador quântico real é a porta quântica que faz a multiplicação por $a^x \, mod \, C$. Já

que temos que implementá-la através de outras portas mais fundamentais e não diretamente através da matriz como na simulação.

Primeiro implementei um circuito mais simples com 5 qubits usando $C = 15$ e $a = 4$ que têm um período de apenas 2 o que permite encolher bastante o circuito.

Para fazer o multiplicador por $a^x \, mod \, 15 = 4^x \, mod \, 15$. Dividimos ele em duas partes: $4^1 \, mod \, 15$ controlado pelo qubit de x menos significativo e $4^2 \, mod \, 15$ controlado pelo mais significativo. Assim se o qubit de x menos significativo for 1 multiplicamos f por $4 \, mod \, 15$ e o mesmo para o mais. Obtendo assim $4^{x_0+x_1} \, mod \, 15$. Como $4^2 \, mod \, 15 = 16 \, mod \, 15 = 1$ precisamos fazer apenas o do bit menos significativo. Para multiplicar o número 1 por 4 sempre que o qubit menos significativo de x for 1 basta usar uma porta CNOT que zera a porta com 1 do digito menos significativo com peso 1 e uma que seta o qubit com peso 4.

Agora executamos o algoritmo com a=2 o que vai requerer 6 qubits. O novo multiplicador é assim: $a^x \, mod \, 15 = 2^x \, mod \, 15$. Dividimos ele em 3 partes: $2^1 \, mod \, 15$ controlado pelo qubit de x menos significativo, $2^2 \, mod \, 15$ controlado pelo qubit do meio e $2^4 \, mod \, 15$ controlado pelo mais significativo. Assim se o qubit de x menos significativo for 1 multiplicamos f por $2 \, mod \, 15$ e o mesmo para os outros. Obtendo assim $4^{x_0+x_1+x_2} \, mod \, 15$. Como $2^4 \, mod \, 15 = 16 \, mod \, 15 = 1$ precisamos fazer apenas o do meio e o do qubit menos significativo. Para multiplicar um número por 2 módulo 15 vamos usar três portas de Fredkin. Para números binários, a multiplicação por dois é um deslocamento para a esquerda. Como fazemos o módulo 15 do valor também note que os números dão a volta pela esquerda de volta a direta, como uma rotação. Para fazer esse deslocamento usamos portas de Fredkin para fazer SWAPs controlados nos qubits de f. Para multiplicar por $2^2 = 4$ o circuito vai funcionar de maneira semelhante. Só que com um deslocamento de dois qubits.

2 Conclusões

Ao longo da realização do projeto, tivemos contato com diversas áreas do conhecimento, relacionado principalmente com a computação quântica. Dentro da própria física, conceitos como a superposição quântica e fases da função de onda foram revisitados para melhor compreensão, como também um estudo aprofundado dos algorítimos de Groover de busca e Shor de fatoração de números inteiros. Desenvolvemos as implementações destes algorítimos em Python e fizemos simulações para diversas situações. Utilizando os computadores quânticos da IBM implementamos os algoritmos de Shor e Groover e comparamos os resultados com nossa simulação clássica, obtendo resultados similares.

Em conclusão, desenvolvemos com sucesso todo o processo de implementação, simulação e análise de algorítimos quânticos, e fizemos um estudo de tópicos avançados de mecânica quântica com enfase em computação quântica.

Referências

[1] D. Candela. Undergraduate computational physics projects on quantum computing. *American Journal of Physics*, 83(8):688–702, 7 2015. ISSN 0002-9505. doi: 10.1119/1.4922296. URL `http://aapt.scitation.org/doi/full/10.1119/1.4922296`.

[2] IBM. IBM Q Experience, 2019. URL `https://quantumexperience.ng.bluemix.net/qx/experience`.