

# Arquitetura de Computadores Quânticos e Implementação de Portas Lógicas Quânticas em Ambiente 3D.

Eduardo de Paula Lima Nascimento

Prof. Dr. Aleardo Manacero Junior

Universidade Estadual Paulista “Júlio de Mesquita Filho”

## RESUMO

A presente pesquisa tem por objetivo levantar aspectos importantes sobre a arquitetura desenvolvida em computadores quânticos, trazendo assim um maior conhecimento na área para os alunos que estão cursando a disciplina Arquitetura de Computadores no programa de mestrado da UNESP. Para um trabalho mais didático foram pesquisados alguns algoritmos que ajudam no aprendizado de computação quântica. A computação quântica em si é uma área muito complexa e de difícil entendimento, desse modo, para passar o conteúdo com mais clareza, foi desenvolvido um algoritmo contendo portas lógicas quânticas dentro de um ambiente 3D.

**Palavras chave:** Computação Quântica; Ambiente 3D; Algoritmos Quânticos.

## ABSTRACT

*The current research has the purpose to raise important aspects about the architecture present in quantum computers, bringing a bigger knowledge in this area for the students that are taken the course Computer Architecture on the Master Course of UNESP. For a more didactic work, algorithms that can help with learning of quantum computing were surveyed. Quantum Computing is a very complex and very difficult to understand area, thereby, to pass the content with more clarity, was developed an algorithm that works with quantum gates on a 3D environment.*

**Keywords:** Quantum Computation; 3D Environment; Quantic Algorithms.

## 1. INTRODUÇÃO –

“Se você acha que entende mecânica quântica, você não entende mecânica quântica”, Richard Feynman usou essa frase para definir mecânica quântica, pela dificuldade de se pesquisar e entender essa área. Depois de várias pesquisas pode-se concluir que essa frase também pode ser aplicada

em computação quântica, que pode ser considerada uma das áreas mais complexas e mais promissoras da computação na atualidade.

Segundo a Lei de Moore, o número de transistores dos *chips* teria aumento de 100% pelo mesmo custo, a cada 18 meses. Essa previsão foi feita na década de 1960 e vale até hoje, porém atualmente existem dificuldades físicas para suprir essa lei, o tamanho dos transistores está cada vez menor, e em algum momento não existirá capacidade física de desenvolver transistores menores, além de que com o aumento do número de transistores aumenta a dissipação de calor, atrapalhando o desenvolvimento dos *chips*.

Com a dificuldade de aumentar o número de transistores, estava na hora de pesquisar alternativas para o aumento da velocidade e armazenamento de dados dos computadores. Dentre muitas alternativas está a Computação Quântica, na teoria ela seria a sucessora dos computadores atuais, usando outras formas de processar e armazenar dados. Em 1950, para representar um *bit* de informação em computadores eram necessários  $10^{19}$  átomos, a previsão para 2020 é que consiga representar esse único *bit* com apenas um átomo [1], ou seja, o limite físico poderá ser alcançado em 2020.

Os computadores atuais, ou computadores clássicos (como chamam os pesquisadores de computadores quânticos), usam como forma de processar dados os chamados *bits*, um *bit* é a menor unidade de informação que pode ser armazenada ou transmitida em um computador. *Bit* quer dizer *binary digit*, dígito binário em português, e pode assumir somente dois valores, por exemplo, 1 e 0, verdadeiro e falso, ligado e desligado. Em computadores, um *bit* pode ser armazenado por cargas elétricas que estão acima ou abaixo de certo padrão, mas também podem ser armazenados de outras maneiras, como polarização magnética, usada por discos rígidos [1].

Tabela 1.1 – Representação de um *bit*.

| 0         | 1          |
|-----------|------------|
| DESLIGADO | LIGADO     |
| FALSO     | VERDADEIRO |

Então, um *bit* pode apresentar apenas dois valores, desse modo, entra em cena o *qubit*. *Qubit* quer dizer *quantum bit*, ele é a essência da computação quântica, um *qubit* pode apresentar o valor 0 do *bit*, o valor 1, e os dois valores ao mesmo tempo, que é o fenômeno da superposição. Existem alguns princípios básicos da computação quântica e a superposição é um deles. Na superposição um *qubit* pode armazenar dois valores binários ao mesmo tempo.

Um *qubit* pode ser imaginado como um átomo que pode tanto atribuir rotação no sentido horário, quanto no sentido anti-horário, esse fenômeno não tem uma explicação sólida por enquanto, mas acontece, portanto, a melhor solução é aceitar e continuar a pesquisa.

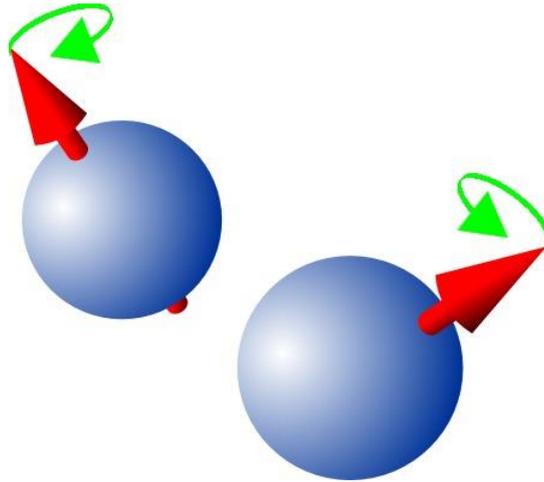


Figura 1.1 – Rotação de Partículas. Fonte: [www.azoquantum.com](http://www.azoquantum.com).

## 2. QUBIT –

Um *qubit* é representado pela nomenclatura  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ , onde  $|\varphi\rangle$  representa o valor do *qubit*,  $\alpha$  representa a amplitude de probabilidade de  $|0\rangle$  acontecer e  $\beta$  representa a amplitude de probabilidade de  $|1\rangle$  acontecer. As amplitudes são normalizadas  $|\alpha|^2 + |\beta|^2 = 1$ . Nesse momento fica complicado o entendimento, mas ao longo da pesquisa, os conceitos vão ficando mais claros.

Na Figura 1.2 pode ser visto um exemplo de *qubit*, onde tem estado  $|0\rangle$  quando está voltado para cima, estado  $|1\rangle$  quando está voltado para baixo, e estado  $|0\rangle$  e  $|1\rangle$  quando está entre os dois estados. Nesse exemplo temos  $\alpha = \frac{1}{\sqrt{2}}$  e  $\beta = \frac{1}{\sqrt{2}}$ .

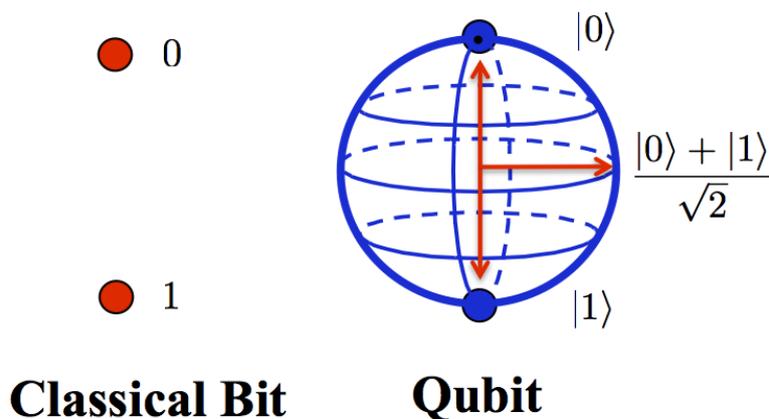


Figura 2.1 – Representação de *bits* clássicos e de *qubit*. Fonte: [qoqms.phys.strath.ac.uk](http://qoqms.phys.strath.ac.uk).

Isso caracteriza o princípio da superposição, onde um elemento pode estar em dois lugares ao mesmo tempo.

Um estado do *qubit* pode ser representado em forma de vetor:

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

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

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

### 3. EMARANHAMENTO (*ENTANGLEMENT*) –

Outro princípio fundamental da computação quântica é o *entanglement* [2], que une dois ou mais *qubits*, resultando todos os estados contidos neles.

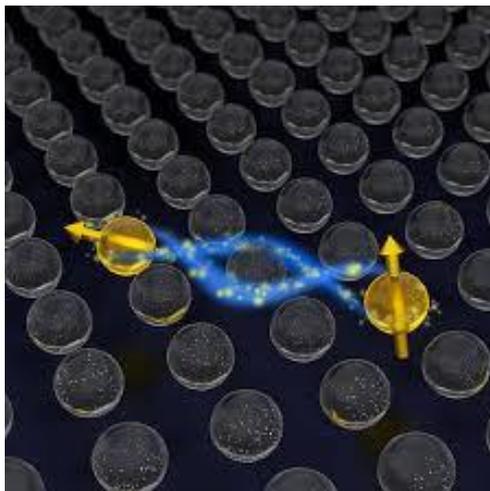


Figura 3.1 – Representação de um *entanglement* entre dois átomos.

Esse fenômeno acontece quando dois *qubits* estão sendo processados em uma mesma operação. As próximas equações mostram como são tratados os *qubits* que sofre *entanglement*.

$$|\varphi_1\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\varphi_2\rangle = \gamma|0\rangle + \delta|1\rangle$$

$$|\varphi\rangle = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

Colocando em um vetor:

$$|\varphi\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}$$

## 4. PORTAS QUÂNTICAS –

Para operar com os *qubits* são usadas portas quânticas, que serão demonstradas nesse capítulo.

Para explicar as portas quânticas, serão explicadas, primeiramente, as portas lógicas usadas em computadores clássicos.

### 4.1 Portas Lógicas

Portas lógicas ou circuitos lógicos, são dispositivos que operam um ou mais sinais lógicos de entrada para produzir uma e somente uma saída, dependente da função implementada no circuito. As portas lógicas mais usadas são: AND, OR e NOT.

A porta AND tem como resultado verdadeiro apenas se as duas entradas forem verdadeiras, sua representação está na Figura 4.1, e a Tabela 4.1 mostra os resultados da porta a partir de diferentes entradas.



Figura 4.1 – Porta AND.

Tabela 4.1 – Porta AND.

| A | B | AND |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

A porta OR tem como resultado verdadeiro se alguma das duas entradas for verdadeira, sua representação está na Figura 4.2, e a Tabela 4.2 mostra os resultados da porta a partir de diferentes entradas.



Figura 4.2 – Porta OR.

Tabela 4.2 – Porta OR.

| A | B | AND |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |

|   |   |   |
|---|---|---|
| 1 | 1 | 1 |
|---|---|---|

A porta NOT troca o valor da entrada, sua representação está na Figura 4.3, e a Tabela 4.3 mostra os resultados da porta a partir de diferentes entradas.

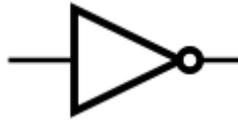


Figura 4.3 – Porta NOT.

Tabela 4.3 – Porta NOT.

| A | AND |
|---|-----|
| 0 | 1   |
| 1 | 0   |

## 4.2 Portas Quânticas

As portas quânticas diferente das portas lógicas clássicas precisam de mais valores de saída, não apenas um, porque com os estados em *entanglement* precisamos medir todos os *qubits*. As portas quânticas são matrizes unitárias, ou seja, elas são reversíveis, uma matriz reversível quer dizer que uma matriz multiplicada pela sua transposta resulta na própria matriz, salvando assim o resultado dos *qubits*.

Existem portas quânticas que operam com apenas um *qubit* e portas quânticas que operam com mais *qubits*.

### 4.2.1 Porta NOT

A porta quântica NOT segue a mesma diretriz da porta NOT clássica, troca o valor de entrada pelo seu oposto, ela é representada pela matriz:

$$NOT = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Quando um *qubit* é operado na porta NOT, temos:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$$

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

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

### 4.2.2 Porta Hadamard

A porta Hadamard transforma estados simples em superposições, ela é representada pela matriz:

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

Ela opera da seguinte forma:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} X \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha + \beta \\ \alpha - \beta \end{bmatrix}$$

#### 4.2.3 Porta C-NOT

Essa porta atua com mais de um *qubit*, ela é uma porta controlada, e altera o resultado do segundo *qubit* se o primeiro for 1.

$$C - NOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Operação:

$$|\varphi\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} X \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix} = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\delta \\ \beta\gamma \end{bmatrix}$$

$$|\varphi_1\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$|\varphi_2\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

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

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

#### 4.2.4 Porta ZC-NOT

Essa porta atua com mais de um *qubit*, ela é uma porta controlada, e altera o resultado do segundo *qubit* se o primeiro for 0.

$$ZC - NOT = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Operação:

$$|\varphi\rangle = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\gamma \\ \beta\delta \end{bmatrix} = \begin{bmatrix} \alpha\gamma \\ \alpha\delta \\ \beta\delta \\ \beta\gamma \end{bmatrix}$$

$$|\varphi_1\rangle = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|\varphi_2\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

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

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

## 5. ALGORITMOS QUÂNTICOS –

Nesse capítulo serão apresentados alguns algoritmos quânticos, usando as portas lógicas vistas anteriormente.

### 5.1 Algoritmo de Deutsch

Esse foi o primeiro algoritmo onde foi mostrado que um computador quântico tem vantagem em um computador clássico. O problema de Deutsch consiste em saber se determinada função  $f(x)$  é constante ou balanceada, uma função constante é quando  $f(x)$  é igual para qualquer  $x$ , e uma função balanceada é quando  $f(x)$  é diferente [3], isso pode ser visto melhor na tabela:

| x | f1(x) | f2(x) | f3(x) | f4(x) |
|---|-------|-------|-------|-------|
| 0 | 1     | 0     | 0     | 1     |
| 1 | 1     | 0     | 1     | 0     |

Para um computador clássico essa operação teria que ser feita duas vezes, testando  $x = 0$  e  $x = 1$ . Para um computador quântico essa operação teria que ser feita apenas uma vez. A Figura 5.1.1 apresenta o circuito feito para resolver o problema de Deutsch.

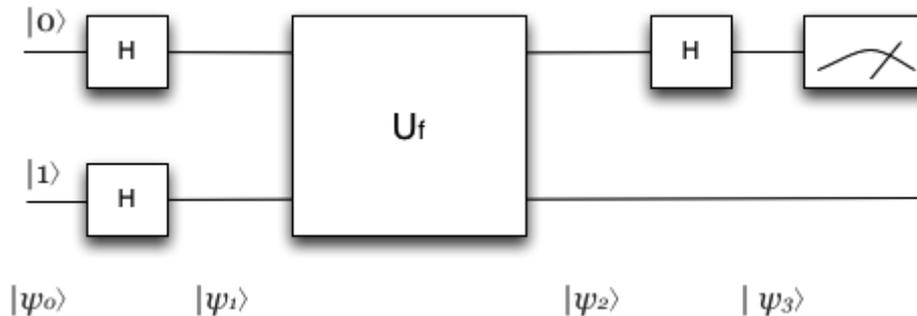


Figura 5.1.1 – Algoritmo de Deutsch

A porta  $U_f$  no circuito é uma matriz unitária e é implementada a partir do resultado da função.

Começando o circuito os dois *qubits* passam por portas Hadamard, para isso acontecer deve ser feito o produto tensor das duas portas, porque nessa fase os *qubits* já estão interligados, ou seja, em estado de *entanglement*.

Temos dois *qubits*:

$$|\varphi_1\rangle = |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$|\varphi_2\rangle = |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

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

A porta Hadamard é:

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

O tensor entre duas portas Hadarmad:

$$H \otimes H = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Fazendo a multiplicação dos *qubits* pela porta:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$$

Esse resultado agora vai para a porta  $U_f$ , pela tabela podemos ver as funções constantes e balaceadas:

| x | f1(x) | f2(x) | f3(x) | f4(x) |
|---|-------|-------|-------|-------|
| 0 | 1     | 0     | 0     | 1     |
| 1 | 1     | 0     | 1     | 0     |

Usando a função  $f1(x)$  como exemplo:

| x | f1(x) |
|---|-------|
| 0 | 1     |
| 1 | 1     |

Podemos ver que quando o  $x = 0$  a função muda o  $x$ . Usando *qubits*:

| <i>qubit</i> | <i>qubit</i> resultante |
|--------------|-------------------------|
| 00>          | 01>                     |
| 01>          | 00>                     |
| 10>          | 10>                     |
| 11>          | 11>                     |

Colocando em matriz:

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

Portanto a  $U_f$  é a porta ZC-NOT com a função  $f1$ . Multiplicando o resultado obtido anteriormente por essa matriz:

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \frac{1}{2} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$$

Por último o resultado passa por uma porta Hadamard apenas e no primeiro *qubit*, para isso acontecer encontra-se o produto tensor dessa porta com uma matriz identidade, pois o segundo *qubit* não será alterado.

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

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

O tensor entre duas portas Hadarmad:

$$H \otimes I = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Multiplicando o resultado obtido anteriormente por essa matriz:

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \times \frac{1}{2} \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} = \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ -2 \\ 2 \end{bmatrix}$$

O resultado então é:

$$|\varphi\rangle = \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix} \begin{bmatrix} 0 \\ 0 \\ \frac{-1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

Nota-se que a probabilidade do primeiro *qubit* ser 1 é 100%, então nossa saída é

$$|\varphi_1\rangle = |1\rangle$$

Se fosse utilizada uma função balanceada, a porta  $U_f$  seria diferente, e a saída seria:

$$|\varphi_1\rangle = |0\rangle$$

Conclui-se que dependendo da função, com apenas uma operação podemos saber se a função é contínua ou balanceada.

## 5.2 Algoritmo de Grover

O Algoritmo de Grover é usado para busca de dados. Atualmente um algoritmo de busca em listas desordenadas possui a complexidade de  $O(n)$ , o algoritmo de Grover consegue realizar buscas em listas desordenadas com complexidade de  $O(\sqrt{n})$ .

O Algoritmo inverte o sinal do *qubit* quando  $f(\text{qubit}) = 1$ , e não altera o *qubit* se  $f(\text{qubit}) = 0$ . O circuito pode ser visto na figura 5.2.1.

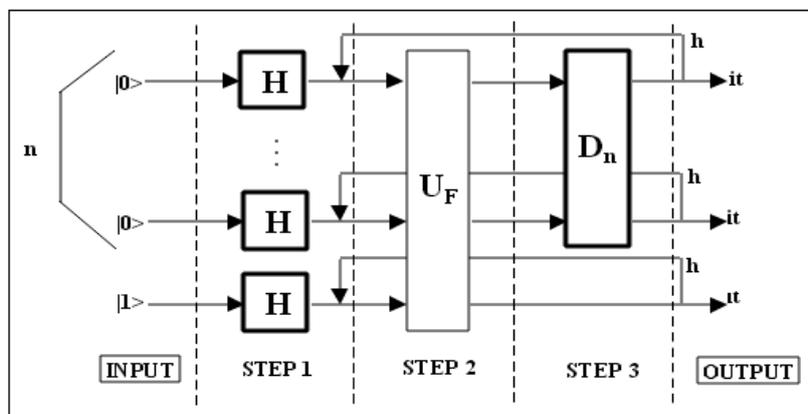


Figura 5.2.1 – Circuito do Algoritmo de Grover. Fonte: [4].

Como no algoritmo anterior  $U_f$  depende da  $f(x)$ , ou seja, depende de qual número a busca tem que retornar. A matriz  $D$  é uma matriz de controle como podemos ver na figura 5.2.2.

| $D_n$          | $ 0..0\rangle$ | $ 0..1\rangle$ | ... | $ i\rangle$    | ... | $ 1..0\rangle$ | $ 1..1\rangle$ |
|----------------|----------------|----------------|-----|----------------|-----|----------------|----------------|
| $ 0..0\rangle$ | $-1+1/2^{n-1}$ | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | $1/2^{n-1}$    |
| $ 0..1\rangle$ | $1/2^{n-1}$    | $-1+1/2^{n-1}$ | ... | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | $1/2^{n-1}$    |
| ...            | ...            | ...            | ... | ...            | ... | ...            | ...            |
| $ i\rangle$    | $1/2^{n-1}$    | $1/2^{n-1}$    | ... | $-1+1/2^{n-1}$ | ... | $1/2^{n-1}$    | $1/2^{n-1}$    |
| ...            | ...            | ...            | ... | ...            | ... | ...            | ...            |
| $ 1..0\rangle$ | $1/2^{n-1}$    | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | ... | $-1+1/2^{n-1}$ | $1/2^{n-1}$    |
| $ 1..1\rangle$ | $1/2^{n-1}$    | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | ... | $1/2^{n-1}$    | $-1+1/2^{n-1}$ |

Figura 5.2.2 – Matriz  $D$ . Fonte: [4].

Usando esse circuito podemos obter o resultado esperado da busca na complexidade  $O(\sqrt{n})$ .

## 6. IMPLEMENTAÇÃO DE PORTAS LÓGICAS EM AMBIENTE 3D

Foi feita uma implementação de portas lógicas em um ambiente 3D para um melhor entendimento de como funciona o processamento de *qubits*. Para a implementação foi escolhido o algoritmo de Deutsch. O ambiente usado é o OpenSim que é uma plataforma de código aberto, a programação foi feita na linguagem *Isl*, uma linguagem de *script* desenvolvida pela *Linden Lab*.

Primeiramente foi construído o circuito do algoritmo, como é visto na figura 6.1.

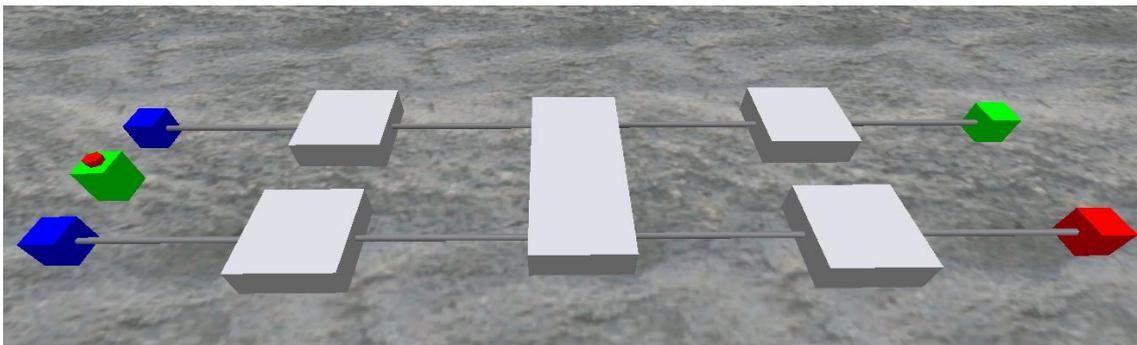


Figura 6.1 – Circuito do Algoritmo de Deutsch

Os dois cubos azuis são as entradas com os *qubits* correspondentes:

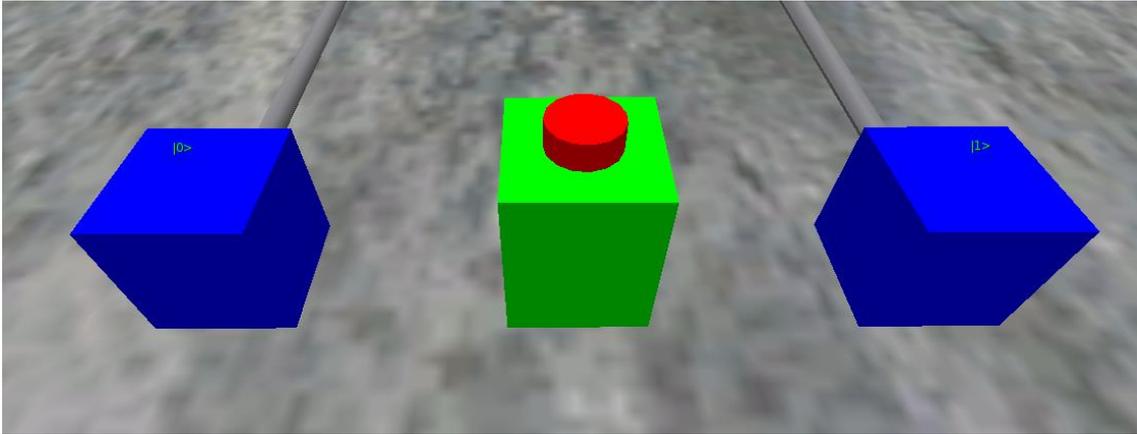


Figura 6.2 – Entrada do circuito do Algoritmo de Deutsch

As duas primeiras portas são portas Hadarmard, e fazem a operação como foi visto anteriormente.

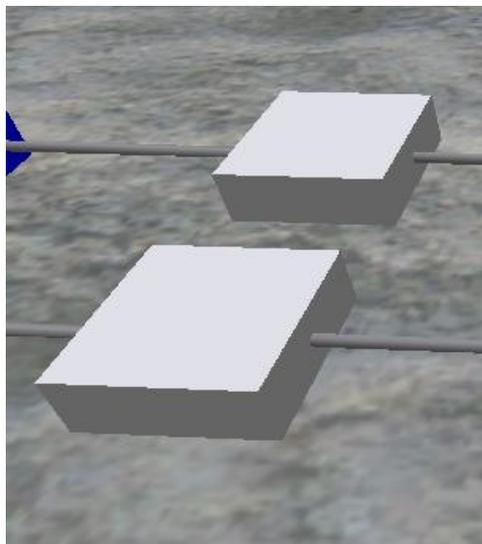


Figura 6.3 – Portas Hadamard

Logo em seguida é a porta  $U_f$ , que muda de acordo com a  $f(x)$  apresentada.

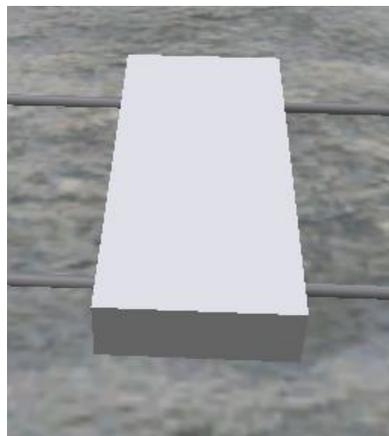


Figura 6.4 – Porta  $U_f$

Para decidir qual porta usar foi feito um método de entrada, que o usuário coloca o valor de  $f(x)$  logo quando inicia o circuito.

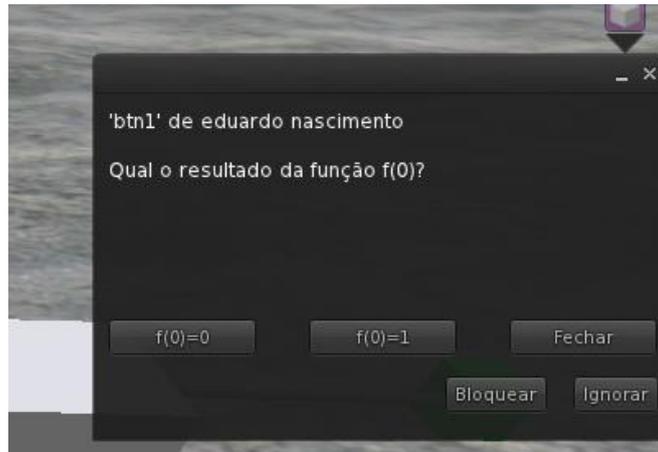


Figura 6.5 – Entrada para  $f(0)$ .

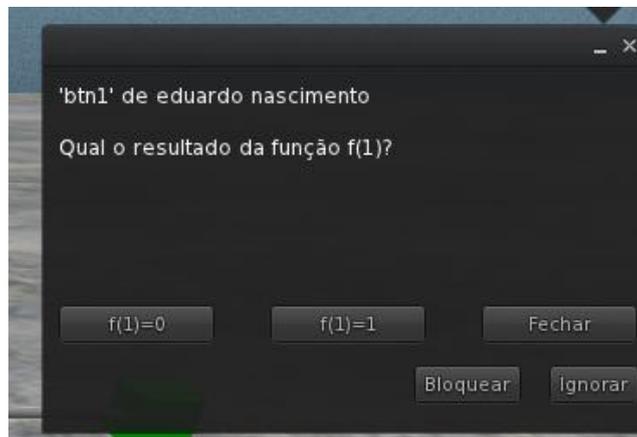


Figura 6.6 – Entrada para  $f(1)$ .

As duas últimas portas são Hadamard e Identidade, respectivamente.



Figura 6.7 – Portas Hadamard e Identidade.

Por último está a saída, lida apenas no primeiro *qubit*.

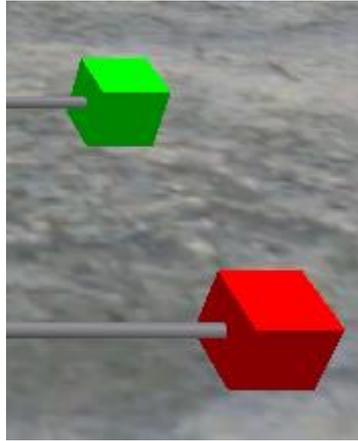


Figura 6.8 – Saídas de circuito.

Para um exemplo de funcionamento, serão aplicados  $f(0) = 1$  e  $f(1) = 0$ , uma função balanceada. Para iniciar basta clicar no botão vermelho, visto na figura 6.2. Quando iniciado irá aparecer a figura 6.5, depois de escolhido “ $f(0) = 1$ ”, aparecerá a figura 6.6, onde será escolhido “ $f(1)=0$ ”, quando escolhido o sistema começa a funcionar, e a passar dados entre as portas.

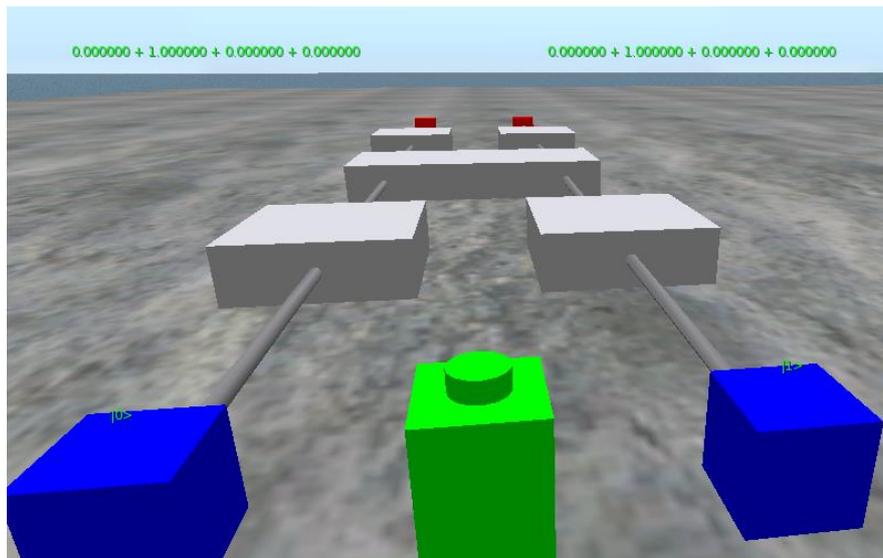


Figura 6.9 – Começando o circuito com o *qubits* em *entanglement*.

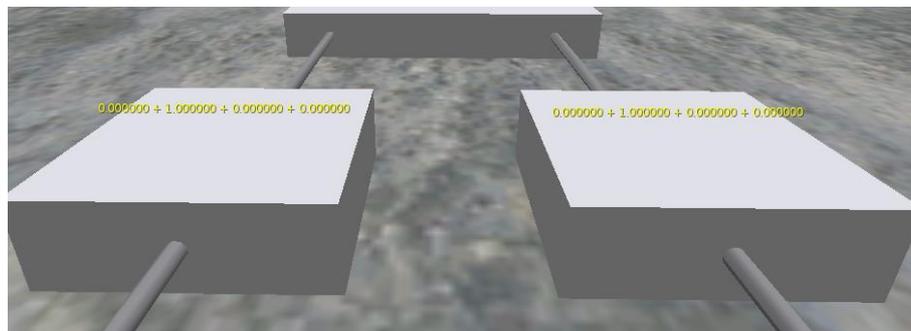


Figura 6.10 – Estado chegando às portas Hadamard

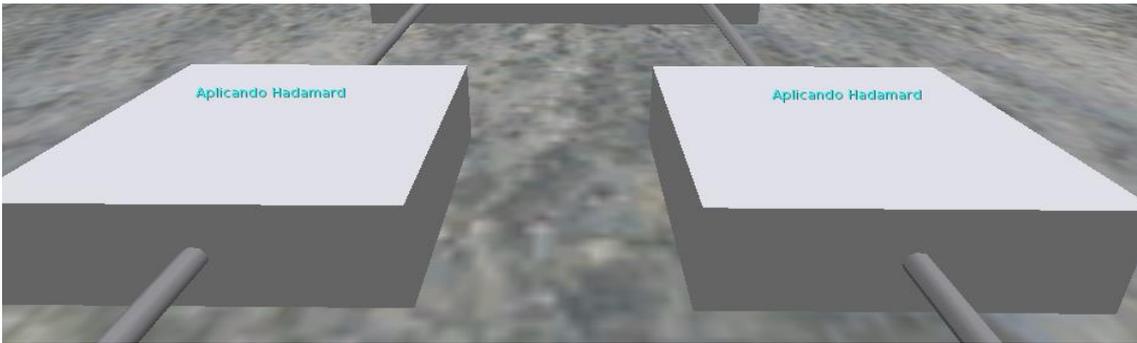


Figura 6.11 – Aplicando Hadamard no estado.

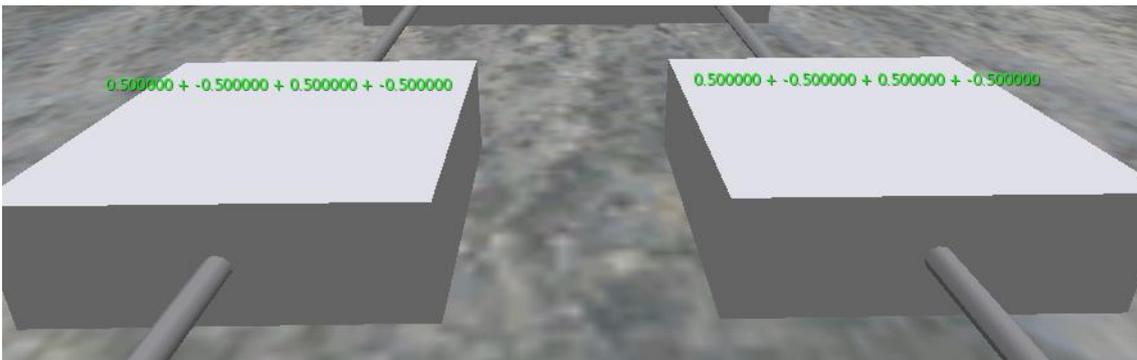


Figura 6.12 – Resultado do Hadamard no estado.

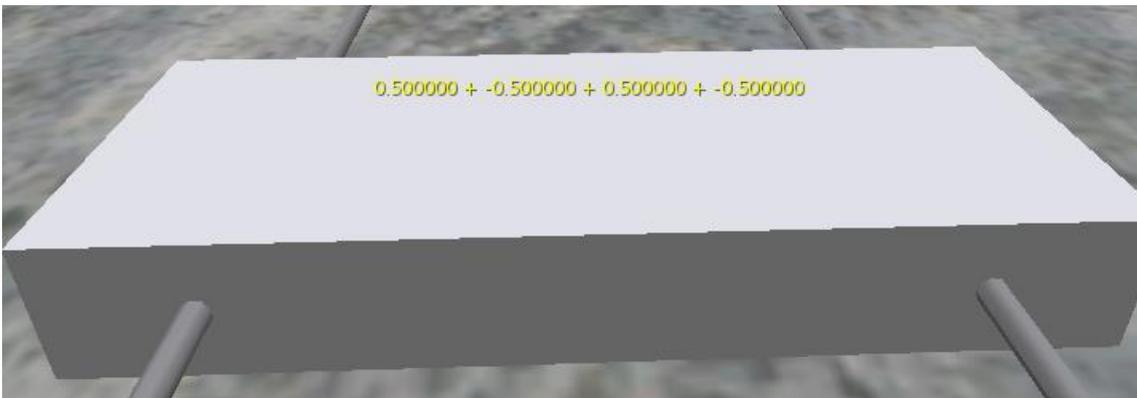


Figura 6.13 – Estado chegando na porta Uf.

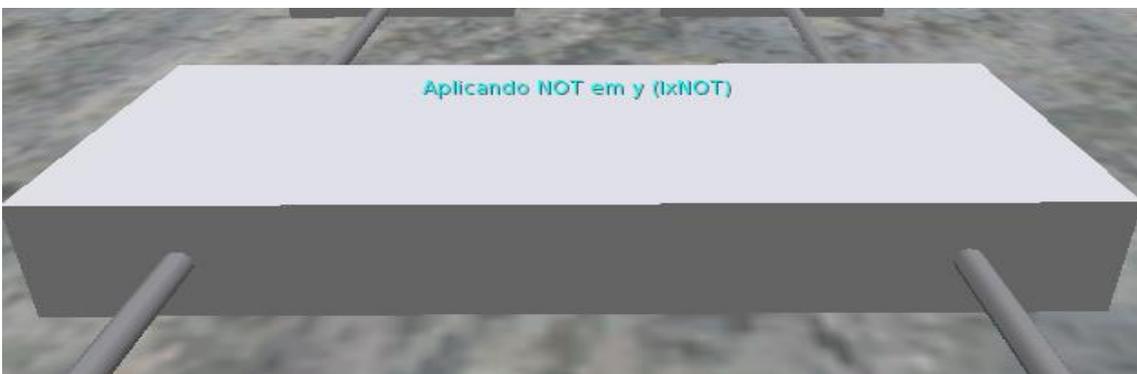


Figura 6.14 – Aplicando a porta IxNOT no estado.

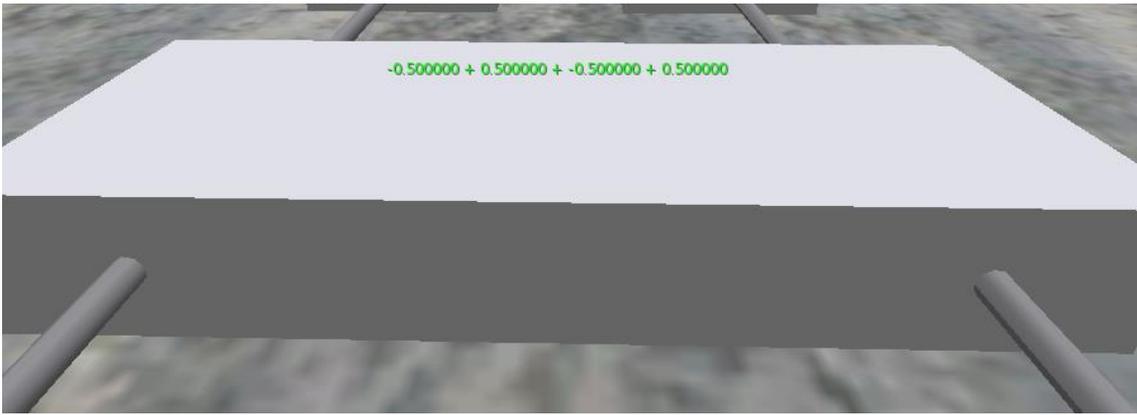


Figura 6.15 – Resultado de IxNOT no estado.



Figura 6.16 – Aplicando a porta HxI no estado.

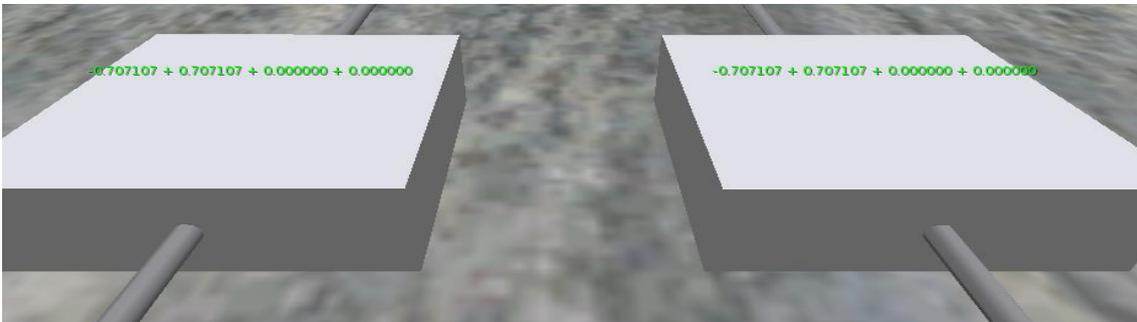


Figura 6.17 – Resultado de HxI no estado.



Figura 6.18 – Saída do Circuito.

Podemos ver por esse exemplo que a saída do primeiro *qubit* é zero, mostrando que a função é balanceada.

## 7. CONSTRUÇÃO DE PROCESSADORES QUÂNTICOS

Existem vários métodos de se construir um computador quântico, porém nem todos são eficientes, o método mais usado atualmente é a construção de *chips* supercondutores, que atuam em uma temperatura muito baixa para operar com os átomos do metal supercondutor.

Em 2011 foi lançado o primeiro computador quântico funcional, o D-Wave. Seu microchip é visto na figura 7.1.

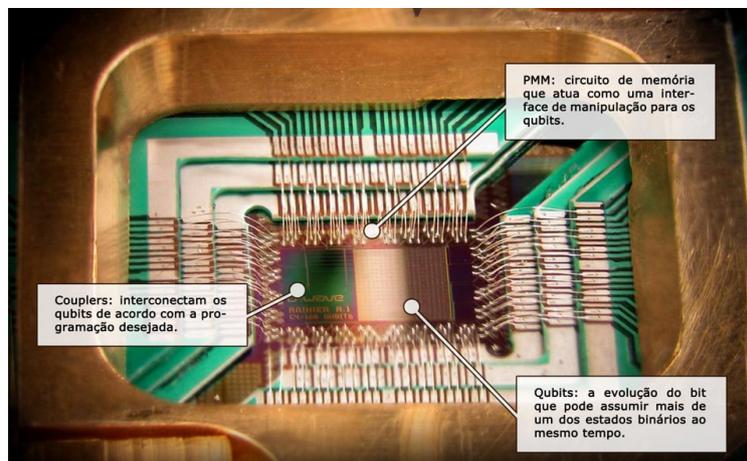


Figura 7.1 – Processador D-Wave. Fonte: <http://www.comppet.ufu.br/>

Esse computador foi construído apenas para pesquisa, isso porque, seu processamento é muito limitado e consegue executar poucas tarefas. O preço desse computador é de U\$\$ 10.000.000,00. Ele possui 128 *qubits* e é mantido em um sistema criogênico, blindado em um espaço de 10m<sup>2</sup>, conseguindo uma temperatura de -272.98 °C.

## 8. REFERÊNCIAS

[1] ARRUDA, L. G. E. *Computação quântica baseada em medidas projetivas em sistemas quânticos abertos*. Tese (Doutorado – Programa de Pós-Graduação em Física – Área de Concentração: Física Básica) – Instituto de Física de São Carlos da Universidade de São Paulo. 2011.

[2] MENDONÇA, P. E. M. F. *Estudos de portas lógicas quânticas de dois qubits definidas em um subespaço livre de decoerência para um sistema de quatro qubits acoplados ao resto do universo por um agente degenerado*. Dissertação

(Mestrado – Programa de Pós-Graduação em Física) – Instituto de Física de São Carlos da Universidade de São Paulo. 2004.

[3] CABRAL, G; LIMA, A; LULA, B. *Interpretando o algoritmo de Deutsch no interferômetro de Mach-Zehnder*. Revista Brasileira de Ensino de Física v.26, n.2, p. 109 - 116, (2004).

[4] SERGUEI V. ULYANOV, FABIO GHISI, SERGUEI A. PANFILOV, VIKTOR S. ULYANOV, ICHIRO KURAWAKI, LUDMILA LITVINTSEVA. *SIMULATION OF QUANTUM ALGORITHMS ON CLASSICAL COMPUTERS*. 1999.