**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.

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:

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

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.

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.

**linguagens quânticas**

**Qiskit (Python-based)**

Desenvolvido pela IBM para programação de computadores quânticos baseados em sua plataforma IBM Quantum.

**Q# (Microsoft)**

* Desenvolvido pela Microsoft como parte do seu projeto Quantum Development Kit (QDK).
* Focado em facilitar a construção de algoritmos quânticos, com integração estreita com o .NET.

**Cirq (Python-based)**

Desenvolvido pelo Google para trabalhar com circuitos quânticos em sua plataforma quantum computing (Google Quantum AI).

**Passos da estrutura quântica** **Programação em Linguagem de Alto Nível:**

O processo começa com a programação do algoritmo quântico, geralmente escrito em linguagens de alto nível como Q#, Cirq, Qiskit, entre outras. Essas linguagens permitem que os desenvolvedores escrevam programas que definem os algoritmos quânticos que serão executados nos computadores quânticos.

**Passos da estrutura quântica Compilação:**

Durante a compilação, o algoritmo quântico é transformado em um circuito quântico. Nessa fase, o circuito pode ser otimizado, e um dos principais ajustes é o mapeamento de qubits lógicos para qubits físicos. Isso é necessário porque os qubits lógicos (aqueles usados na programação) podem não corresponder diretamente aos qubits físicos (aqueles que realmente existem na máquina), e a otimização ajuda a garantir que o circuito seja executado de forma eficiente.

**Passos da estrutura quântica Representação Intermediária:**

Após a compilação, o circuito quântico pode ser representado de forma intermediária, o que permite novas manipulações e ajustes. Essa representação pode ser usada para otimizar ainda mais o circuito ou para fazer ajustes em portas quânticas.

**Passos da estrutura quântica Camada de Hardware e Controle:**

Em seguida, a representação intermediária é convertida para instruções específicas do hardware quântico. A camada de hardware é responsável por traduzir as operações do circuito para comandos que podem ser executados fisicamente. Isso inclui a conversão das portas quânticas para as instruções compatíveis com a microarquitetura do hardware, como aqueles que envolvem controle de qubits, calibração e manipulação de estados quânticos.

**Passos da estrutura quântica Comandos de Controle Quântico:**

Os comandos de controle quântico gerados na etapa anterior são responsáveis por enviar sinais precisos para manipular os qubits. Esses sinais podem incluir pulsos de micro-ondas, sinais eletromagnéticos ou pulsos de laser, dependendo da tecnologia de qubits utilizada. O controle quântico, por sua vez, é uma camada transparente para o desenvolvedor, cuidando da calibração dos dispositivos e da execução física das operações quânticas.