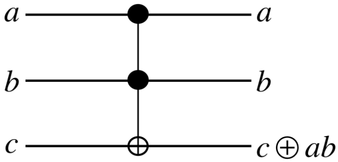
COL863: Quantum Computation and Information

Ragesh Jaiswal, CSE, IIT Delhi

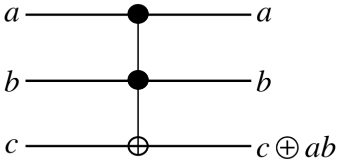
Introduction: Quantum Algorithms

Introduction
Quantum algorithms

- Can we simulate classical logic circuit using a quantum circuit?
- Claim: Any classical logic circuit can be implemented using just `NAND` and `COPY` gates.
- If we can build a quantum analogue of `NAND` and `COPY` gates, then we will be done.
- The following three-qubit gate, called the Toffoli gate, can be used to implement both `NAND` and `COPY`.



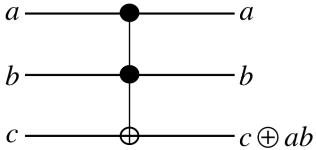
- Can we simulate classical logic circuit using a quantum circuit?
- Claim: Any classical logic circuit can be implemented using just NAND and COPY gates.
- If we can build a quantum analogue of NAND and COPY gates, then we will be done.
- The following three-qubit gate, called the Toffoli gate, can be used to implement both NAND and COPY.



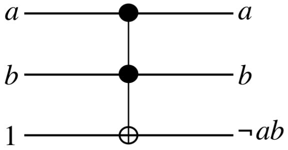
- Question: Can you build NAND using Toffoli gate?

Introduction
Quantum algorithms

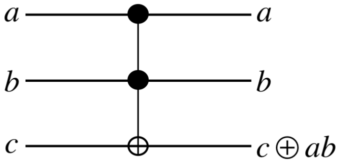
- Can we simulate classical logic circuit using a quantum circuit?
- Claim: Any classical logic circuit can be implemented using just NAND and COPY gates.
- If we can build a quantum analogue of NAND and COPY gates, then we will be done.
- The following three-qubit gate, called the Toffoli gate, can be used to implement both NAND and COPY.



- Question: Can you build NAND using Toffoli gate?

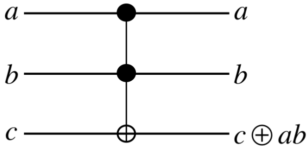


- Can we simulate classical logic circuit using a quantum circuit?
- Claim: Any classical logic circuit can be implemented using just `NAND` and `COPY` gates.
- If we can build a quantum analogue of `NAND` and `COPY` gates, then we will be done.
- The following three-qubit gate, called the Toffoli gate, can be used to implement both `NAND` and `COPY`.

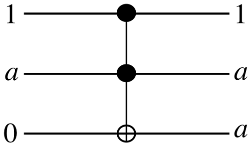


- Question: Can you build `COPY` using Toffoli gate?

- Can we simulate classical logic circuit using a quantum circuit?
- <u>Claim</u>: Any classical logic circuit can be implemented using just `NAND` and `COPY` gates.
- If we can build a quantum analogue of `NAND` and `COPY` gates, then we will be done.
- The following three-qubit gate, called the Toffoli gate, can be used to implement both `NAND` and `COPY`.

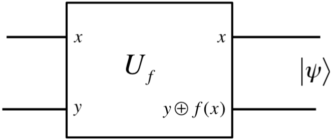


- <u>Question</u>: Can you build `NAND` using Toffoli gate?

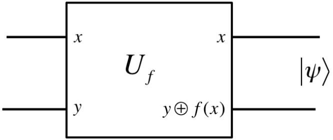


- Can we simulate classical logic circuit using a quantum circuit?
  Yes
- Can quantum circuits do more than just simulating classical ones?
  - We will introduce the idea of quantum parallelism. The main idea is simultaneous evaluation of a function over various inputs.
  - We will look at Deutsch's Algorithm which is a prototypical example used to demonstrate the idea of quantum parallelism.

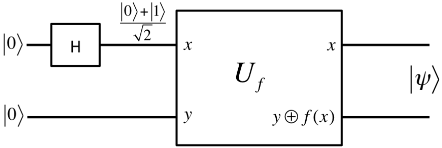
- Consider any boolean function over one-bit inputs $f : \{0,1\} \to \{0,1\}$.
- Claim: It is possible to construct the following quantum gate U_f (using basic gates):



- Consider any boolean function over one-bit inputs $f : \{0,1\} \rightarrow \{0,1\}$.
- Claim: It is possible to construct the following quantum gate U_f (using basic gates):



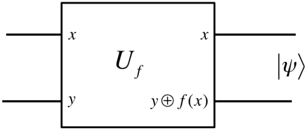
- By feeding inputs $|00\rangle$ and $|10\rangle$, we can compute $f(0)$ and $f(1)$.
- What happens when we feed the input $|+\rangle|0\rangle$ in this circuit? What is the output state $|\psi\rangle$?



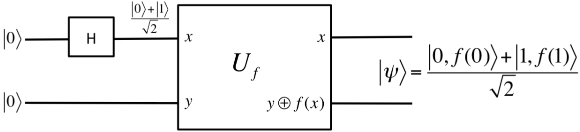
Introduction

Quantum algorithms → Deutsch's algorithm

- Consider any boolean function over one-bit inputs $f : \{0,1\} \to \{0,1\}$.
- <u>Claim</u>: It is possible to construct the following quantum gate U_f (using basic gates):



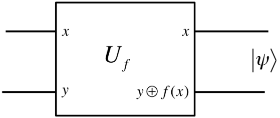
- By feeding inputs $|00\rangle$ and $|10\rangle$, we can compute $f(0)$ and $f(1)$.
- What happens when we feed the input $|\beta_{00}\rangle\,|0\rangle$ in this circuit? What is the output state $|\psi\rangle$?



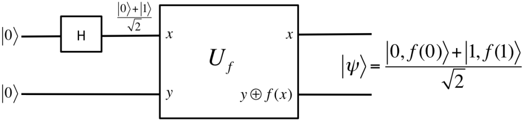
- This output state contains simultaneous evaluations of both $f(0)$ and $f(1)$!

Introduction
Quantum algorithms → Deutsch's algorithm

- Consider any boolean function over one-bit inputs $f : \{0,1\} \rightarrow \{0,1\}$.
- Claim: It is possible to construct the following quantum gate U_f (using basic gates):

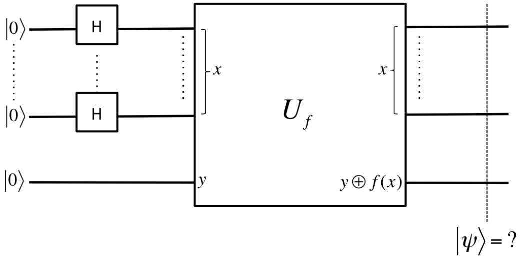


- By feeding inputs $|00\rangle$ and $|10\rangle$, we can compute $f(0)$ and $f(1)$.
- What happens when we feed the input $|\beta_{00}\rangle\,|0\rangle$ in this circuit? What is the output state $|\psi\rangle$?

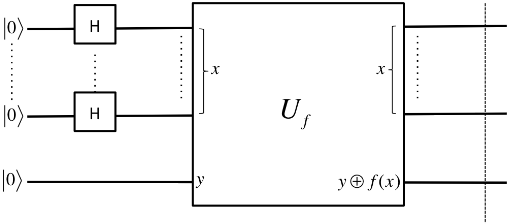


- This output state contains simultaneous evaluations of both $f(0)$ and $f(1)$!
- Question: Can we generalize this idea for boolean functions over multiple bit inputs?

- Question: Can we generalize this idea for boolean functions over multiple bit inputs?
- Consider any boolean function over n-bit inputs $f : \{0,1\}^n \rightarrow \{0,1\}$.
- What is the output of the following circuit?

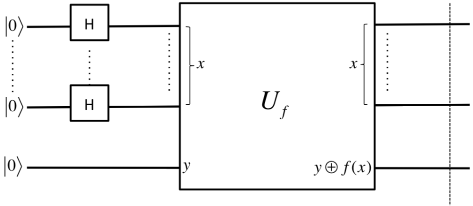


- Question: Can we generalize this idea for boolean functions over multiple bit inputs?
- Consider any boolean function over n-bit inputs $f : \{0,1\}^n \rightarrow \{0,1\}$.
- What is the output of the following circuit?



$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_x |x\rangle |f(x)\rangle$$

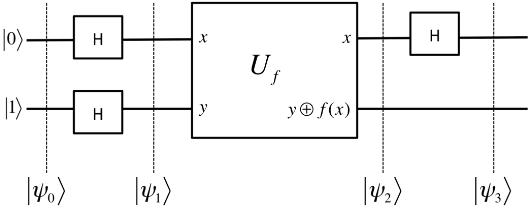
- Question: Can we generalize this idea for boolean functions over multiple bit inputs?
- Consider any boolean function over n-bit inputs $f : \{0,1\}^n \rightarrow \{0,1\}$.
- What is the output of the following circuit?



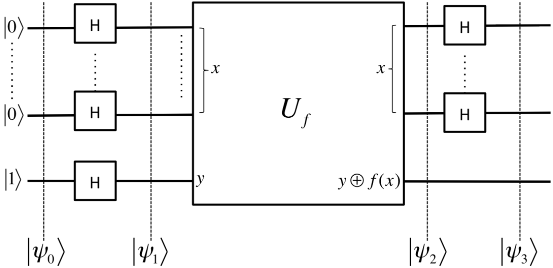
$$|\psi\rangle = \frac{1}{2^{n/2}} \sum_x |x\rangle |f(x)\rangle$$

- Even though final state encodes evaluation of the function on all inputs, what we can measure is only one of them. So, it is important that we do not get carried away by the potential quantum parallelism exhibited in the above circuit.

- Even though final state encodes evaluation of the function on all inputs, what we can measure is only one of them. So, it is important that we do not get carried away by the potential quantum parallelism exhibited in the above circuit.
- Exploiting the parallelism in more realistic way is the key challenge while designing quantum algorithms.
- Consider the case of a boolean function on single-bit inputs $f : \{0, 1\} \rightarrow \{0, 1\}$. Suppose we would want to know if $f(0) = f(1)$. Here is a quantum circuit that solves this.



- The previous problem was a specific case of the more general Deutsch's problem that further demonstrates the power of quantum algorithms.
- Deutsch's problem: Bob has a function $f : \{0,1\}^n \rightarrow \{0,1\}$ that is either a constant function or a balanced function (i.e., f is 0 on $2^n/2$ inputs). Alice wants to determine what kind of function Bob has but can make a query to the function only once.
- The following circuit does this:



End