Quantum Computing

Overview

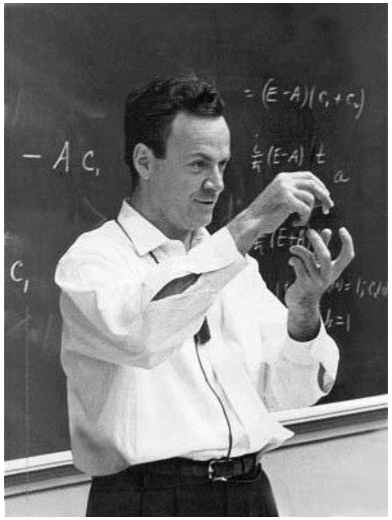
- Introduction and History
- Data Representation
- Operations on Data
- Shor’s Algorithm
- Conclusion and Open Questions

Introduction

What is a quantum computer?

- A quantum computer is a machine that performs calculations based on the laws of quantum mechanics, which is the behavior of particles at the sub-atomic level.

Introduction



- "I think I can safely say that nobody understands quantum mechanics" - Feynman
- 1982 - Feynman proposed the idea of creating machines based on the laws of quantum mechanics instead of the laws of classical physics.
- 1985 - David Deutsch developed the quantum turing machine, showing that quantum circuits are universal.
- 1994 - Peter Shor came up with a quantum algorithm to factor very large numbers in polynomial time.
- 1997 - Lov Grover develops a quantum search algorithm with $O(\sqrt{N})$ complexity

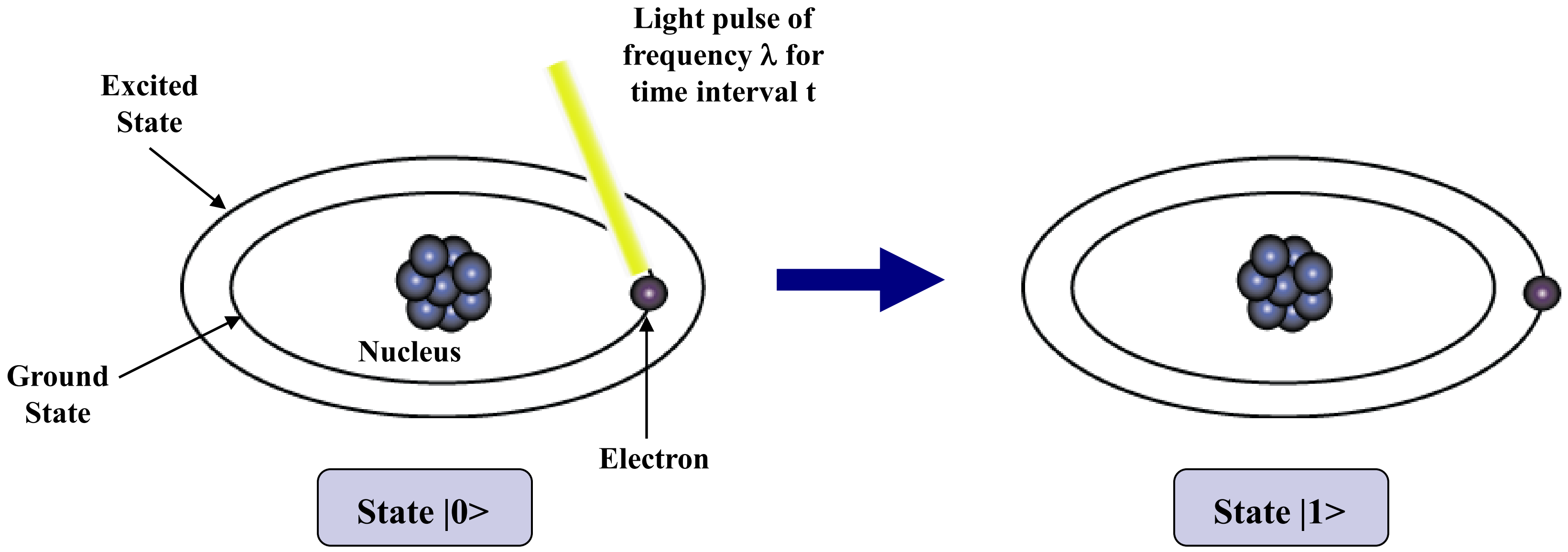
Overview

- ✓ Introduction and History
- **Data Representation**
- Operations on Data
- Shor’s Algorithm
- Conclusion and Open Questions

Representation of Data - Qubits

A bit of data is represented by a single atom that is in one of two states denoted by |0> and |1>. A single bit of this form is known as a ***qubit***

A physical implementation of a qubit could use the two energy levels of an atom. An excited state representing |1> and a ground state representing |0>.



Representation of Data - Superposition

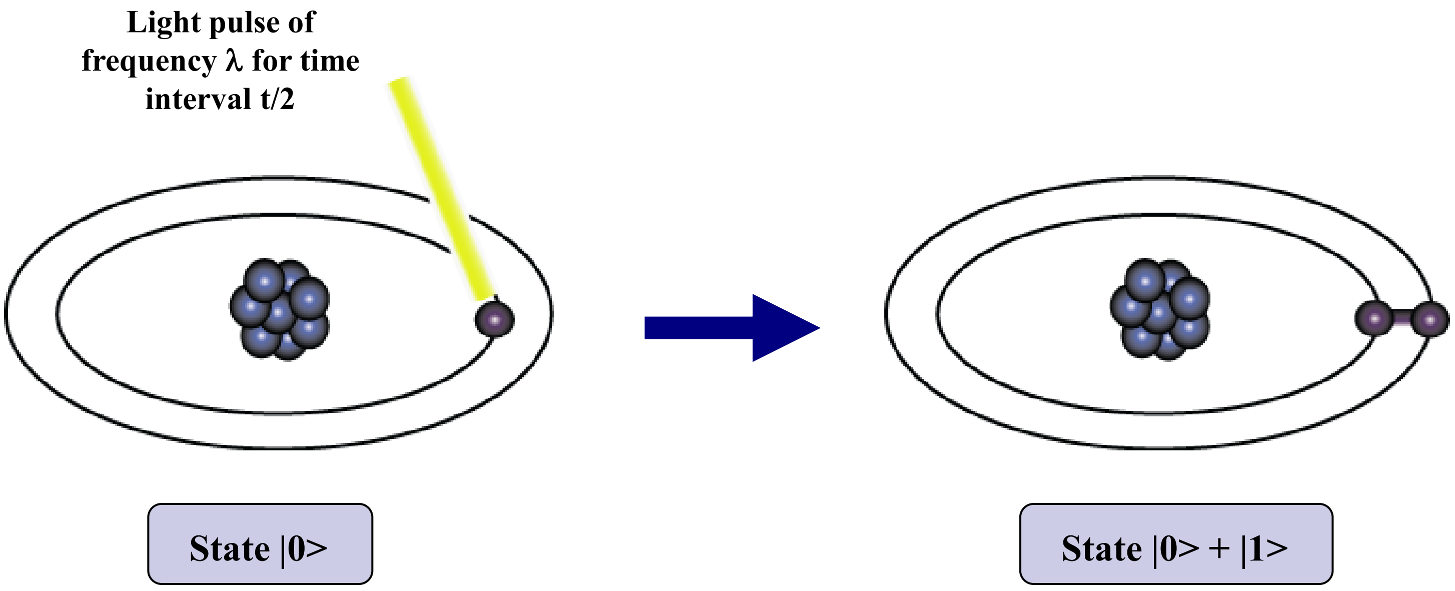
A single qubit can be forced into a ***superposition*** of the two states denoted by the addition of the state vectors:

$$|\psi> = \alpha_1 |0> + \alpha_2 |1>$$

Where α_1 **and** α_2 are complex numbers and $|\alpha_1|^2 + |\alpha_2|^2 = 1$

A qubit in superposition is in both of the states |1> and |0 at the same time

Representation of Data - Superposition



Light pulse of frequency λ for time interval t/2

State |0>

State |0> + |1>

- Consider a 3 bit qubit register. An equally weighted superposition of all possible states would be denoted by:

$$|\psi> = \frac{1}{\sqrt{8}}|000> + \frac{1}{\sqrt{8}}|001> + \ldots + \frac{1}{\sqrt{8}}|111>$$

Data Retrieval

- In general, an n qubit register can represent the numbers 0 through 2^n-1 simultaneously.

Sound too good to be true?…It is!

- If we attempt to retrieve the values represented within a superposition, the **superposition randomly collapses** to represent just one of the original values.

In our equation: $|\psi> = \alpha_1|0> + \alpha_2|1>$, α_1 represents the probability of the superposition collapsing to $|0>$. The α’s are called probability amplitudes. In a balanced superposition, $\alpha = 1/\sqrt{2}^{\,n}$ where n is the number of qubits.

Relationships among data - Entanglement

- ***Entanglement*** is the ability of quantum systems to exhibit correlations between states within a superposition.
- Imagine two qubits, each in the state |0> + |1> (a superposition of the 0 and 1.) We can entangle the two qubits such that the measurement of one qubit is always correlated to the measurement of the other qubit.

Overview

✓ Introduction and History

✓ Data Representation

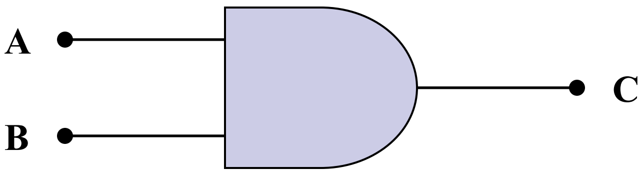
- **Operations on Data**
- Shor’s Algorithm
- Conclusion and Open Questions

Operations on Qubits - Reversible Logic

- Due to the nature of quantum physics, the destruction of information in a gate will cause heat to be evolved which can destroy the superposition of qubits.

Ex.

The AND Gate



Input		Output
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

In these 3 cases, information is being destroyed

- This type of gate cannot be used. We must use ***Quantum Gates***.

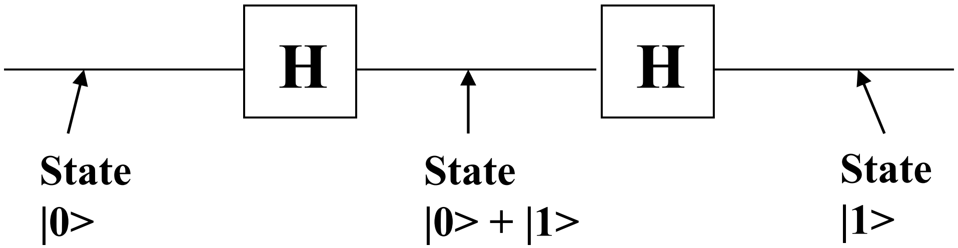
Quantum Gates

- Quantum Gates are similar to classical gates, but do not have a degenerate output. i.e. their original input state can be derived from their output state, uniquely. ***They must be reversible.***

- This means that a deterministic computation can be performed on a quantum computer only if it is reversible. Luckily, it has been shown that any deterministic computation can be made reversible.(Charles Bennet, 1973)

Quantum Gates - Hadamard

- Simplest gate involves one qubit and is called a ***Hadamard Gate (***also known as a square-root of NOT gate.) Used to put qubits into superposition.



State |0>

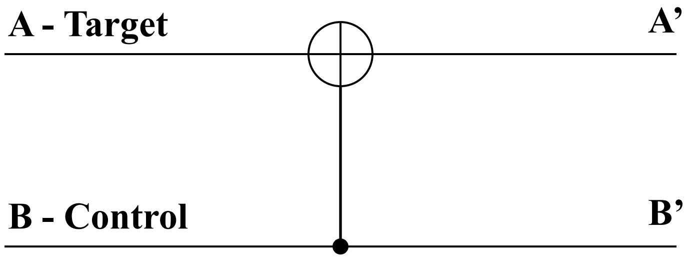
State |0> + |1>

State |1>

Note: Two Hadamard gates used in succession can be used as a NOT gate

Quantum Gates - Controlled NOT

- A gate which operates on two qubits is called a ***Controlled-NOT (CN) Gate.*** If the bit on the control line is 1, invert the bit on the target line.



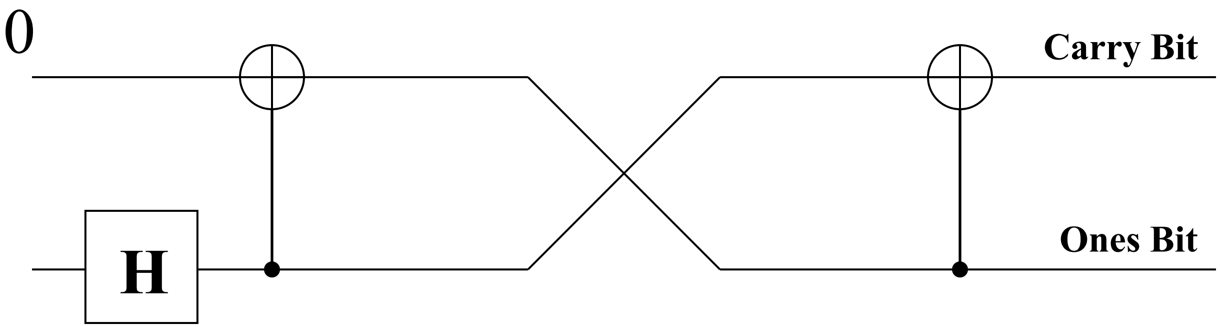
Input		Output	
A	**B**	**A'**	**B'**
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Note: The CN gate has a similar behavior to the XOR gate with some extra information to make it reversible.

Example Operation - Multiplication By 2

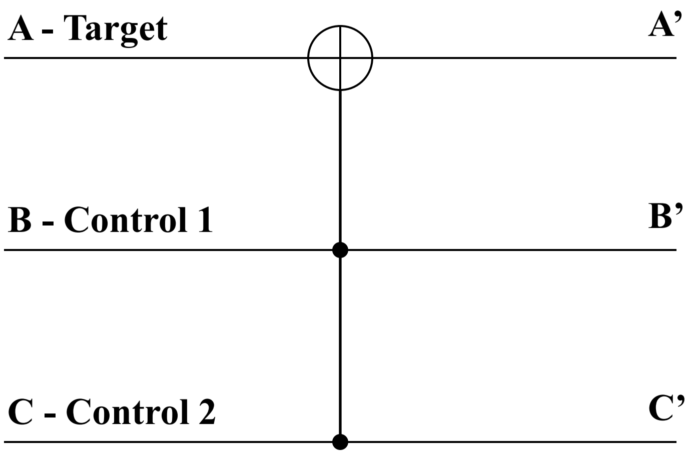
- We can build a reversible logic circuit to calculate multiplication by 2 using CN gates arranged in the following manner:

Input		Output	
Carry Bit	Ones Bit	Carry Bit	Ones Bit
0	0	0	0
0	1	1	0



Quantum Gates - Controlled Controlled NOT (CCN)

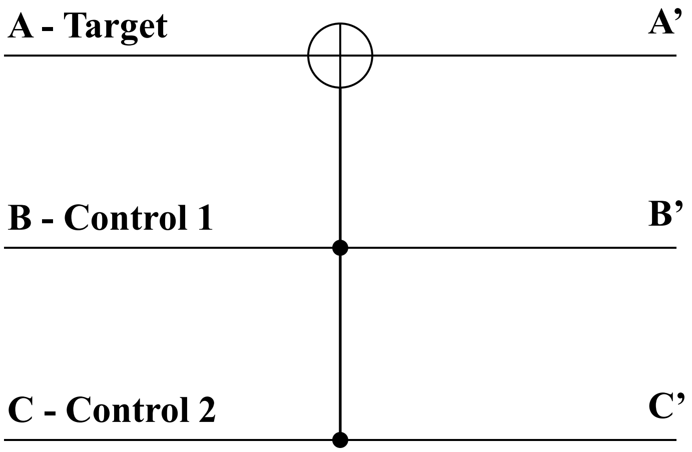
- A gate which operates on three qubits is called a ***Controlled Controlled NOT (CCN) Gate.*** Iff the bits on both of the control lines is 1,then the target bit is inverted.



Input			Output		
A	B	C	A’	B’	C’
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

A Universal Quantum Computer

- The CCN gate has been shown to be a ***universal*** reversible logic gate as it can be used as a NAND gate.



Input			Output		
A	**B**	**C**	**A’**	**B’**	**C’**
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

When our target input is 1, our target output is a result of a NAND of B and C.

Overview

- ✓ Introduction and History
- ✓ Data Representation
- ✓ Operations on Data
- **Shor’s Algorithm**
- Conclusion and Open Questions

Shor’s Algorithm

- Shor’s algorithm shows (in principle,) that a quantum computer is capable of factoring very large numbers in polynomial time.

The algorithm is dependant on

- Modular Arithmetic
- Quantum Parallelism
- Quantum Fourier Transform

Shor's Algorithm - Periodicity

- An important result from Number Theory:

  $\mathbf{F(a) = x^{a} mod\ N}$ is a periodic function

- Choose N = 15 and x = 7 and we get the following:

$$7^{0} \bmod 15 = 1$$

$$7^{1} \bmod 15 = 7$$

$$7^{2} \bmod 15 = 4$$

$$7^{3} \bmod 15 = 13$$

$$7^{4} \bmod 15 = 1$$

$$\vdots$$

Shor's Algorithm - In Depth Analysis

To Factor an odd integer N (Let's choose 15) :

1. Choose an integer q such that $N^2 < q < 2N^2$ **let's pick 256**
2. Choose a random integer x such that GCD(x, N) = 1 **let's pick 7**
3. Create two quantum registers (these registers must also be entangled so that the collapse of the input register corresponds to the collapse of the output register)
   - Input register: must contain enough qubits to represent numbers as large as q-1. **up to 255, so we need 8 qubits**
   - Output register: must contain enough qubits to represent numbers as large as N-1. **up to 14, so we need 4 qubits**

Shor’s Algorithm - Preparing Data

4. Load the input register with an equally weighted superposition of all integers from 0 to *q*-1. **0 to 255**

5. Load the output register with all zeros.

The total state of the system at this point will be:

$$\underbrace{\frac{1}{\sqrt{256}}\sum_{a=0}^{255}}_{\text{Input Register}}|a,\ \underbrace{000}_{\text{Output Register}}>$$

Note: the comma here denotes that the registers are entangled

Shor’s Algorithm - Modular Arithmetic

6. Apply the transformation x^a mod N to each number in the input register, storing the result of each computation in the output register.

Note that we are using decimal numbers here only for simplicity.

Input Register	7^a Mod 15	Output Register
|0>	7^0 Mod 15	1
|1>	7^1 Mod 15	7
|2>	7^2 Mod 15	4
|3>	7^3 Mod 15	13
|4>	7^4 Mod 15	1
|5>	7^5 Mod 15	7
|6>	7^6 Mod 15	4
|7>	7^7 Mod 15	13

⋮

Shor's Algorithm - Superposition Collapse

7. Now take a measurement on the output register. This will collapse the superposition to represent ***just one*** of the results of the transformation, let's call this value *c*.

Our output register will collapse to represent one of the following:

|1>, |4>, |7>, or |13

For sake of example, lets choose |1>

Shor's Algorithm - Entanglement

Now things really get interesting !

8. Since the two registers are entangled, measuring the output register will have the effect of partially collapsing the input register into an **equal superposition** of each state between 0 and q-1 that yielded c (the value of the collapsed output register.)

Since the output register collapsed to |1>, the input register will partially collapse to:

$$\frac{\mathbf{1}}{\sqrt{\mathit{64}}}\ \mathbf{|0>} + \frac{\mathbf{1}}{\sqrt{\mathit{64}}}\ \mathbf{|4>} + \frac{\mathbf{1}}{\sqrt{\mathit{64}}}\ \mathbf{|8>} + \frac{\mathbf{1}}{\sqrt{\mathit{64}}}\ \mathbf{|12>, \ldots}$$

The probabilities in this case are $\frac{\mathbf{1}}{\sqrt{\mathit{64}}}$ since our register is now in an equal superposition of 64 values (0, 4, 8, ... 252)

Shor's Algorithm - QFT

We now apply the Quantum Fourier transform on the partially collapsed input register. The fourier transform has the effect of taking a state |a> and transforming it into a state given by:

$$\frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} |c> * e^{2\pi iac / q}$$

Shor’s Algorithm - QFT

$$\frac{1}{\sqrt{64}} \sum_{a \in A} |a> , |1> \longrightarrow \frac{1}{\sqrt{256}} \sum_{c=0}^{255} |c> * e^{2\pi i a c / 256}$$

Note: A is the set of all values that 7 ^{a} mod 15 yielded 1. In our case A = {0, 4, 8, …, 252}

So the final state of the input register after the QFT is:

$$\frac{1}{\sqrt{64}} \sum_{a \in A} \frac{1}{\sqrt{256}} \sum_{c=0}^{255} |c> * e^{2\pi i a c / 256}, \; |1>$$

Shor’s Algorithm - QFT

The QFT will essentially peak the probability amplitudes at integer multiples of *q*/4 in our case 256/4, or 64.

|0>, |64>, |128>, |192>, …

So we no longer have an equal superposition of states, the probability amplitudes of the above states are now higher than the other states in our register. We measure the register, and it will collapse with high probability to one of these multiples of 64, let’s call this value p.

With our knowledge of q, and p, there are methods of calculating the period (one method is the continuous fraction expansion of the ratio between q and p.)

Shor's Algorithm - The Factors :)

10. Now that the we have the period, the factors of N can be determined by taking the greatest common divisor of N with respect to *x ^ (P/2)* + 1 and *x ^ (P/2)* - 1. The idea here is that this computation will be done on a classical computer.

We compute:

Gcd($7^{4/2}$ + 1, 15) = **5**

Gcd($7^{4/2}$ - 1, 15) = **3**

We have successfully factored 15!

Shor's Algorithm - Problems

- The QFT comes up short and reveals the wrong period. This probability is actually dependant on your choice of *q*. The larger the q, the higher the probability of finding the correct probability.
- The period of the series ends up being odd

If either of these cases occur, we go back to the beginning and pick a new x.

Overview

- ✓ Introduction and History
- ✓ Data Representation
- ✓ Operations on Data
- ✓ Shor’s Algorithm
- ▪ Conclusion and Open Questions

Conclusion

- In 2001, a 7 qubit machine was built and programmed to run Shor’s algorithm to successfully factor 15.
- What algorithms will be discovered next?
- Can quantum computers solve NP Complete problems in polynomial time?