An Introduction to Quantum Computing

July 2025 – Rev 6 – Added appendix describing the classical computer's simulation of a quantum computer.

Table of Contents

1. Introduction

This introduction to quantum computing is intended for everyone, and especially those who have no knowledge of this relatively new technology.

This discussion will be as simple as possible.

A quantum computer can process a particular type of information much faster than can a 'classical' computer.

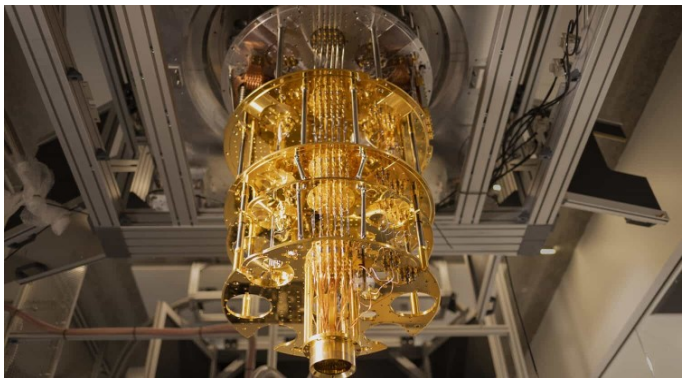
Large companies including Google, Microsoft, IBM and Intel are spending a lot of money and devoting lots of resouces in the development of quantum computers and related software and applications.

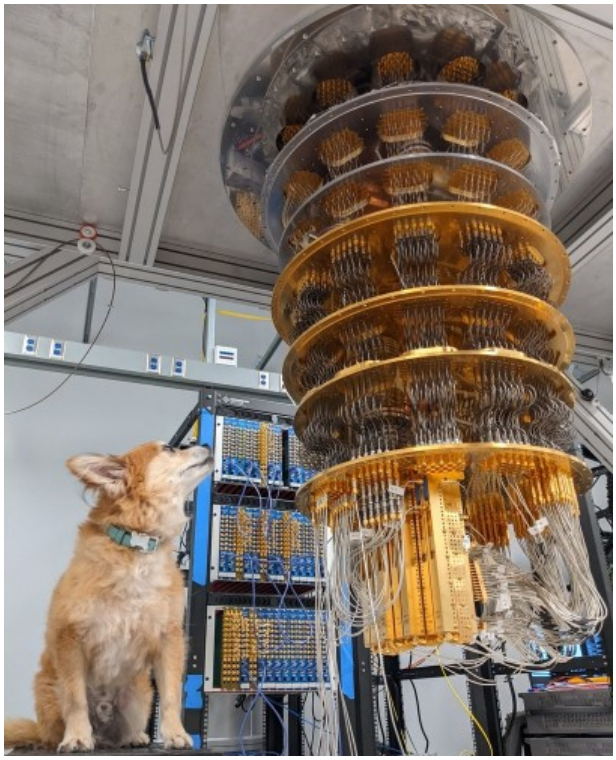
2. Pictures of Classical Computers

Here are pictures of some classical computers. They all work the same way in how they process information. The 'supercomputer' at the bottom is much faster and more expensive than the others.

3. Pictures of Quantum Computers

These are quantum computers from IBM, Google and Microsoft. The dog's name is Qubit.

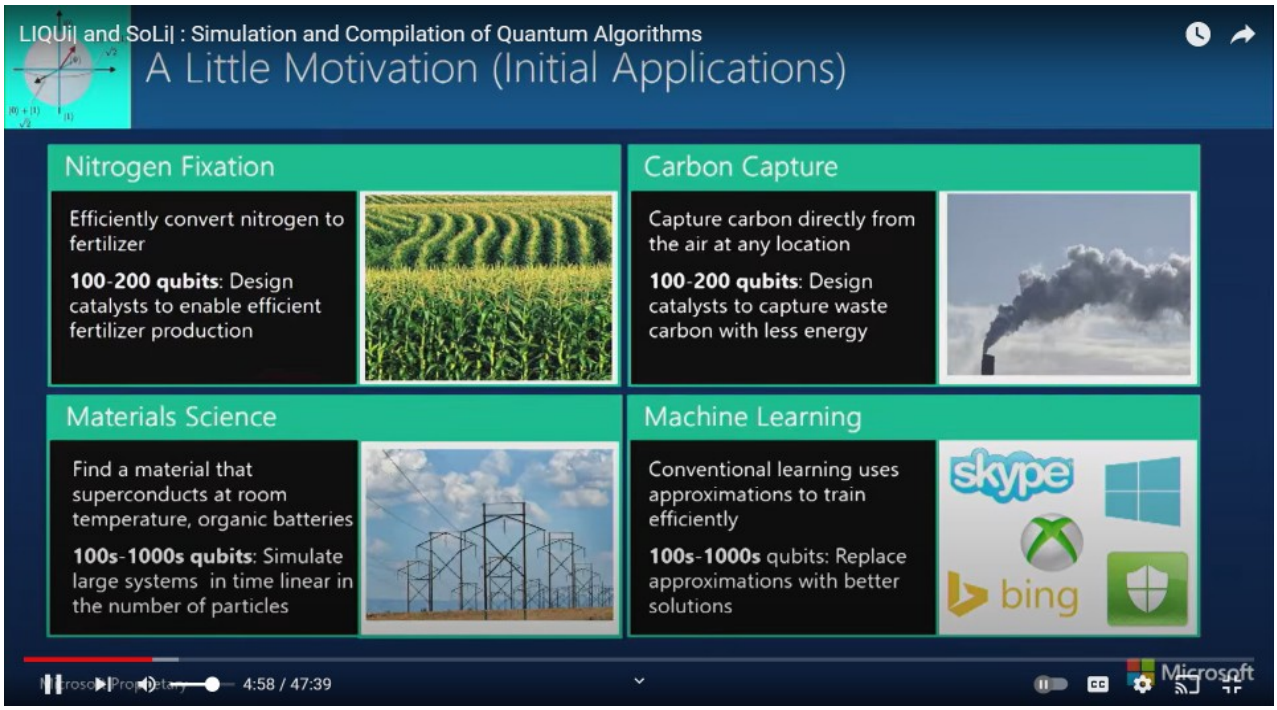




4. An Earth-changing application for Quantum Computers

There are certain tasks that cannot be computed by classical machines because it would take way too long for them to finish.

Creating an efficient way to remove carbon from the atmosphere is a potential Earth-changing application for quantum computers (note 1).



5. Probabilities and information for a single coin

With a single coin there are two pieces of information associated with it. The two pieces of information will indicate the coin's probability of being measured as HEADS or being measured as TAILS.

We can 'measure' the coin by stopping it from spinning and then looking at it, or we can simply look at the coin if it's not spinning.

First we are going to place the coin into an initial state. Here this initialized coin will always be equal to HEADS after we measure it.

For this initialized coin there is a probability of 100% that HEADS will be measured. There is a 0% chance that it will be measured as TAILS. We will write both amounts of probability followed by the resulting states of the coin like this:

100/100|HEADS> or 1|HEADS>
  0/100|TAILS> or 0|TAILS>

Now we are going to spin the coin.

When we measure the spinning coin it will result in the coin being in either the HEADS or the TAILS state with an equal probability.

Just like the initialized coin there are two pieces of information associated with it. In this case, the two pieces of information are now:

50/100|HEADS> or 1/2|HEADS>
50/100|TAILS> or 1/2|TAILS>

The spins/measurements will get closer to being 50% HEADS and 50% TAILS the more we spin, measure and tabulate the results.

6. Probabilities and information for three coins

It's time to work with three coins.

Since there are two pieces of information associated with a single coin, it would seem that there are six pieces of information associated with these three coins. However, there is another way of looking at the information contained in these three coins.

When considering the coins in combination there are eight pieces of information associated with three coins. These eight pieces of information reflect the probabilities of measuring the three coins in these states:

|HEADS HEADS HEADS>
|HEADS HEADS TAILS>
|HEADS TAILS HEADS>
|HEADS TAILS TAILS>
|TAILS HEADS HEADS>
|TAILS HEADS TAILS>
|TAILS TAILS HEADS>
|TAILS TAILS TAILS>

First, the three coins will be placed into their initialized state.

When the three coins are measured they will all be HEADS.

The eight probabilities associated with these three initialized coins are:

1|HEADS HEADS HEADS> All three coins will always measure HEADS
0|HEADS HEADS TAILS>
0|HEADS TAILS HEADS>
0|HEADS TAILS TAILS>
0|TAILS HEADS HEADS>
0|TAILS HEADS TAILS>
0|TAILS TAILS HEADS>
0|TAILS TAILS TAILS>

Let's put the three coins into their spinning states. Now all eight of the states of the three coins will have equal probabilities of one-out-of-eight.

1/8|HEADS HEADS HEADS>
1/8|HEADS HEADS TAILS>
1/8|HEADS TAILS HEADS>
1/8|HEADS TAILS TAILS>
1/8|TAILS HEADS HEADS>
1/8|TAILS HEADS TAILS>
1/8|TAILS TAILS HEADS>
1/8|TAILS TAILS TAILS>

7. Probabilities and information for one-hundred coins

In general, the number of pieces of information for any given number of coins is:

= $2^{\text{number of coins}}$

That is, multiply the number 2 together as many times as you have coins.

Let's imagine that we have 100 coins. The number of pieces of information associated with these 100 coins is:

= 2^{100} pieces of information for 100 coins

= 2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*
2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*
2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*
2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*
2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2*2 pieces of information for 100 coins

= (approximately) 1,000,000,000,000,000,000,000,000,000 pieces of information for 100 coins

= one-million-trillion-trillion pieces of information for 100 coins

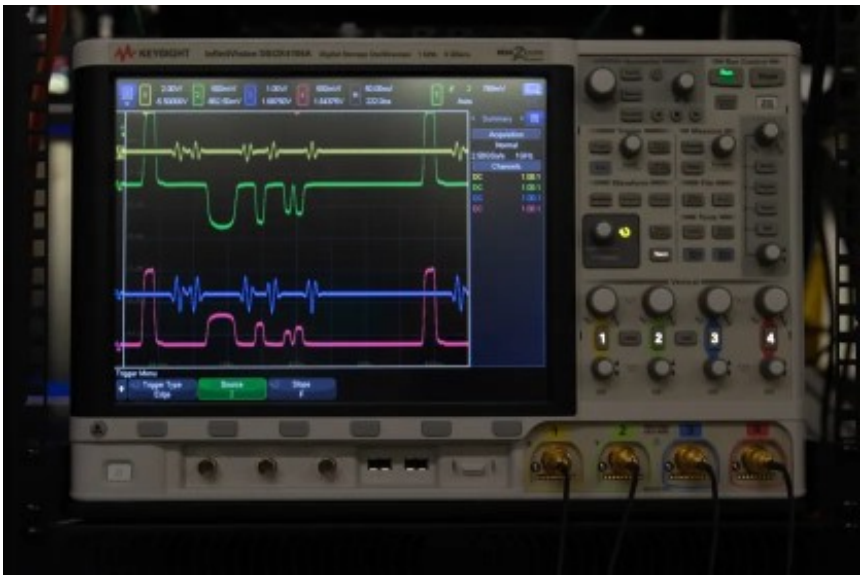
This is obviously a lot of information.

8. Quantum binary digits (qubits)

Quantum computers use quantum binary digits or qubits.

Qubits are 'zapped' by the user in order to modify and then measure their states (note 2). Measuring each qubit reveals one of its two possible values to us.

These pictures show a quantum computer being programmed, and an oscilloscope display showing the waveforms of the microwave energy that is zapping the qubits.



9. Qubits and information

Similar to how we described the operation done on a coin, a qubit when measured will result in it having one of two values.

Associated with the combinations of qubits are also probability amplitudes associated with what the measured values of the individual qubits might be.

Each of the possible combinations for the qubits is called a 'basis state'.

Unlike coins, however, the values of the qubits' probability amplitudes are progressively modified by the user of a quantum computer. This continues until the user decides to measure the qubits in order to reveal a meaningful answer from the qubits' basis states.

10. Modifying qubit information

The time required to zap qubits and modify all of their associated basis state probability amplitudes is very fast (note 3).

On the other hand, if a classical supercomputer is used to modify similar amounts of probability information it can take a very long time.

This table compares how long it might take a quantum computer and a classical supercomputer to modify the same amount of probability information (see Appendix A).

The number of pieces of information for the indicated number of qubits	**Time needed for a quantum computer to modify this many pieces of information**	**Time needed for a classical supercomputer to modify this many pieces of information**
Two pieces of information are contained in 1 qubit	0.000001 seconds (one microsecond)	0.00000000000000006 seconds (much faster than a quantum computer for 1 qubit of information)
One-million pieces of information are contained in 20 qubits	0.000001 seconds (one microsecond)	0.000001 seconds (the same amount of time as a quantum computer for 20 qubits)
One-thousand-trillion pieces of information are contained in 50 qubits	0.000001 seconds (one microsecond)	40,000 years (the electricity bill will be enormous)
One-million-trillion-trillion pieces of information are contained in 100 qubits	0.000001 seconds (one microsecond)	50,000-million-trillion-trillion years (our universe would be long gone by then)

11. Grover's algorithm

Here is a simple example of how the algorithm known as Grover's algorithm might operate on a quantum computer.

Grover's algorithm can be used for searching.

First we are going to consider a Grover's algorithm that is searching through 16 envelopes (aka basis states).
15 of the 16 envelopes each has a worthless small green piece of paper inside.

1 of the 16 envelopes contains a one-thousand-dollar bill.

The algorithm works by successively zapping four qubits in order that the probability associated with one of the sixteen possible basis states becomes much larger than the other fifteen basis states. The basis state that ends up with the highest probability is the envelope with the prize.

This table shows how the sixteen basis states of the four qubits change from the initialized state, then to the equal-probability state, and then through four generations of basis state probability updates (note 4).

The two possible measured states for each qubit will be written as:
|u>
|d>

The sixteen basis states will range from:
|uuuu> through |dddd>

Notice that in Generation 4 the probability amount for one of the sixteen basis states ends up being equal to 1. This is the envelope with the money since all passes through the algorithm and then measurement of the the four qubits will always yield the |duuu> basis state.

Grover's algorithm	Initialized qubits	Equal probability state (one zapping)	Generation 1 (six more zappings)	Generation 2 (six more zappings)	Generation 3 (six more zappings)	Generation 4 (six more zappings)
Envelope 1	1|uuuu>	1/16|uuuu>	6/100|uuuu>	3/100|uuuu>	1/100|uuuu>	0|uuuu>
Envelope 2	0|uuud>	1/16|uuud>	6/100|uuud>	3/100|uuud>	1/100|uuud>	0|uuud>
Envelope 3	0|uudu>	1/16|uudu>	6/100|uudu>	3/100|uudu>	1/100|uudu>	0|uudu>
Envelope 4	0|uudd>	1/16|uudd>	6/100|uudd>	3/100|uudd>	1/100|uudd>	0|uudd>
Envelope 5	0|uduu>	1/16|uduu>	6/100|uduu>	3/100|uduu>	1/100|uduu>	0|uduu>
Envelope 6	0|udud>	1/16|udud>	6/100|udud>	3/100|udud>	1/100|udud>	0|udud>
Envelope 7	0|uddu>	1/16|uddu>	6/100|uddu>	3/100|uddu>	1/100|uddu>	0|uddu>
Envelope 8	0|uddd>	1/16|uddd>	6/100|uddd>	3/100|uddd>	1/100|uddd>	0|uddd>
Envelope 9	0|duuu>	1/16|duuu>	10/100|duuu>	55/100|duuu>	85/100|duuu>	**1|duuu>**
Envelope 10	0|duud>	1/16|duud>	6/100|duud>	3/100|duud>	1/100|duud>	0|duud>
Envelope 11	0|dudu>	1/16|dudu>	6/100|dudu>	3/100|dudu>	1/100|dudu>	0|dudu>
Envelope 12	0|dudd>	1/16|dudd>	6/100|dudd>	3/100|dudd>	1/100|dudd>	0|dudd>
Envelope 13	0|dduu>	1/16|dduu>	6/100|dduu>	3/100|dduu>	1/100|dduu>	0|dduu>
Envelope 14	0|ddud>	1/16|ddud>	6/100|ddud>	3/100|ddud>	1/100|ddud>	0|ddud>
Envelope 15	0|dddu>	1/16|dddu>	6/100|dddu>	3/100|dddu>	1/100|dddu>	0|dddu>
Envelope 16	0|dddd>	1/16|dddd>	6/100|dddd>	3/100|dddd>	1/100|dddd>	0|dddd>

12. Shor's algorithm

We will conclude with a brief discussion of Shor's algorithm. It was created by Peter Shor in 1994. Its main feature is that it can factor very large number much faster when run on a quantum computer than on a classical computer. Since its creation in 1994, Shor's algorithm has raised awareness for the potential of quantum computing.

For the one-digit number '6' it is easy to find its two prime factors.

For the two-digit number '15' it is also very easy to factor.

The three-digit number '143' might take a fourth-grade student a couple of minutes to find the two factors '11' and '13'.

A number with six-hundred digits is effectively impossible for classical supercomputers to factor because it would take them trillions of years to find the two factors.

The RSA and Diffie-Hellman encryption schemes are what keep our internet transactions secure because they utilize a technique that requires the factoring of a six-hundred digit number (2048 bits) in order to break the encryption.

A large enough quantum computer (6,000 error-corrected qubits) will be able factor a six-hundred digit number in less than an hour.

We are many years away from having a quantum computer large enough to threaten our online data security. There are also quantum encryption schemes being developed that will keep us safe. Quantum encryption is way ahead of classical-encryption breaking.

13. Notes and other resources

Notes:

1) Link to video discussing carbon capture (at 3min50s) - https://www.youtube.com/watch?v=4mMizLpIVKs
2) 'Zapping' and 'measuring' certain types of qubits involves exposing the qubits to precise amounts of microwave electromagnetic radiation.
3) Zapping a single qubit or even multiple qubits will probably be around one microsecond. For small quantum computers this is currently faster, but when large amounts of qubits become available then multiplexing and demultiplexing of the zapping waveforms will likely be used.
4) The specific Grover's algorithm used in simulation for the values shown in the table is from Fig.1d here: https://www.nature.com/articles/s41467-017-01904-7

Other resources:

The Sounds of IBM - IBM :
https://www.youtube.com/watch?v=o-FyH2A7Ed0

Inside the Google Quantum AI Campus - Google :
https://www.youtube.com/watch?v=2uV5XwhH6Eg

The Map of Quantum Computing - Domain of Science :
https://www.youtube.com/watch?v=-UlxHPIEVqA

14. Appendix A – Simulation of a quantum computer

A classical computer can simulate the operation of a quantum computer using its multiply-add operation. To do this, the classical computer will apply the matrices of quantum gates onto the state vector of the qubits.

A quantum computer starts out with the state vector for each of its qubits as follows:

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

This indicates that the probability amplitude for each qubit is such that there is a 100% chance (the top value) of measuring each qubit in the state '0', and a 0% chance (the bottom value) of measuring them in the state '1'.

To change the probability amplitude of a qubit, the state vector for the qubit must be multiplied by the matrix assocatiated with the desired quantum gate.

A Hadamard gate may be used to modify the initial states of one of these qubits into a state that has an equal probability of being measure in the '0' and '1' state.

$$1 / \sqrt{(2)} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 1 / \sqrt{(2)} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For a single un-entangled qubit, it only takes 4 multiply-add operations of the simulating classical computer to modify the state vector for this qubit.

Upon initialization, a quantum computer has all of its qubits un-entangled. If there are 100 qubits, it will only take 400 multiply-add operations to modify the states of all of these un-entangled qubits. This will be very fast for any modern classical computer.

However, in a quantum computer, the two-qubit Controlled NOT (CNOT) gate is often used. This usually has the effect of entangling the involved qubits.

Once qubits are entangled with other qubits, the state vectors for those entangled qubits are affected by the others involved in this entanglement. This means that a quantum gate cannot be applied separately to individual qubits. Rather, the associated matrices for the quantum gates must combined using the 'tensor product' (also known as the 'Kronecker product'), and then this combined matrix must then be applied to the state vector for the entangled qubits.

The state vector for a 100 qubit quantum computer, with all of its qubits entangled, is 2^{100} in length.

The size of the associated quantum gates' matrix will be 2^{100} by 2^{100} .

The number of multiply-add operations that need to be performed by the simulating classical computer is 2^{200} ≈ 1.6 x 10^{60} .

The fastest supercomputer in 2025 can perform around 10^{18} multiply-add operations per second.

The time needed by a classical supercomputer to update a state vector for 100 entangled qubits would be:

1.6 x 10^{60} multiply-add operations / 10^{18} multiply-add operatlons per second =

1.6 x 10^{42} seconds ≈

5 x 10^{34} years