Quantum Information Theory: MAT 4953 XLS 5983. Lesson 01 (Aug 23-25), Fall 2022.

José A. Morales E., UTSA Math and Physics & A. Depts.

Abstract

An overview of quantum computing and information.

1. Introduction to Classical Computing

1.1. Charles Babbage (1791-1871): the father of the computer

Babbage designed and began the building of mechanical computing machines such as the difference machine and the analytical engine, which were finished by his son (Henry) after Charles's death.

Their architechture was similar to modern computers. There was a separation between data and program memory, the operation was instruction-based, the control unit was able to make conditional jumps, and it had a separate input/output unit.

1.2. Ada Lovelace: the first computer programmer

She was in communication with Babbage. Ada Lovelace wrote one of the first algorithms for Babbage's machine, designed to compute Bernoulli numbers.

1.3. Alan Turing (1912-1954): father of computer science

As it is well known, Turing was a mathematician who worked as a code breaker during WWII (Enigma code history). He is also considered a father of Artificial Intelligence by means of his Turing's test: when in a "conversation between a human and a computer" (think of writing to a chatbot), if a human cannot distinguish what the computer says from what a human would say.

Turing proposed a simple mathematical model for a universal digital computer: the "Turing machine" (1936), which can perform any algorithm and can simulate any other computing device. The basic form of a Turing machine is made of four elements:

1. Tape: for data storage, acting as a memory. It has a sequence of spaces, where each space has on it one of a finite set of symbols.

Email address: jose.morales4@utsa.edu (José A. Morales E., UTSA Math and Physics & A. Depts.)

2. Processor: controls the operations of the machine and it has a finite number of internal states.

3. Finite instruction set: determines the action of the processor, depending both on the tape symbol and the internal state of the processor.

4. Tape head: able to read a symbol on the tape. If instructed to do so by the processor, it can erase this symbol replacing it with another one (write on tape).

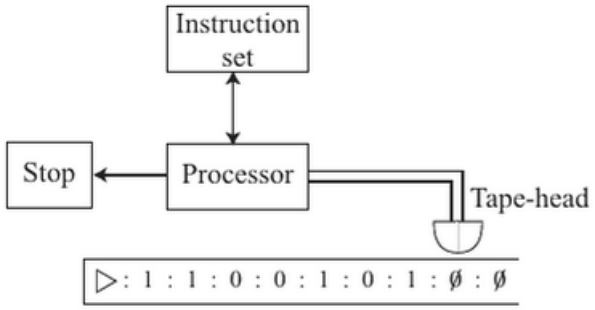


Fig. 7.1 Schematic representation of a Turing machine.

We can move the head along the tape forward or backward if instructed. See Fig. above from the textbook [1] for a representation of the arrangement of these 4 elements. In one of the spaces there's a start marker ($\triangleright$) on the tape. The processor has an initial/start configuration (S) and a final/stop (F) configuration, which once reached then the computation is complete, the processor reporting this to the user. The required value will be on a section of the tape. Notice the difference above between 0 (zero) and $\emptyset$ (blank space).

Example 1. of operation of a Turing machine.

Possible symbols: $\{\triangleright,0,1,\emptyset\}$. Digits 0 and 1 are related to the input data.

Four possible states of the processor: {S,I,II,F}. S: start configuration. F: final state.

Instruction set: We select it to be

$$
\begin{aligned}
(S,\triangleright) &\implies (I,\triangleright),\\
(I,0) &\implies (II,0),\\
\\
(I,1) &\implies (II,1),\\
(II,0) &\implies (F,1),\\
(II,1) &\implies (II,0),
\end{aligned}
$$

$$(II, \emptyset) \implies (I, 1),$$
$$(I, \emptyset) \implies (F, \emptyset).$$

The instructions above include a "move to the next space" (to the right) on the tape.

Case 1. If the tape is prepared in the state

$$\triangleright : 1 : 0 : 1 : \emptyset : \emptyset : \cdots$$

then the steps in the program are (moving on one square after each instruction)

$$(S, \triangleright) \implies (I, \triangleright)[\text{move}](I, 1), \quad (I, 1) \implies (II, 1)[\text{move}](II, 0), \quad (II, 0) \implies (F, 1)$$

and the tape configuration is at this point

$$\triangleright : 1 : 1 : 1 : \emptyset : \emptyset : \cdots$$

Case 2. The tape is prepared now in the state

$$\triangleright : 0 : 1 : 1 : \emptyset : \emptyset : \cdots$$

and the steps in the program are now

$$(S, \triangleright) \implies (I, \triangleright),$$
$$(I, 0) \implies (II, 0),$$
$$(II, 1) \implies (II, 0),$$
$$(II, 1) \implies (II, 0),$$
$$(II, \emptyset) \implies (I, 1),$$
$$(I, \emptyset) \implies (F, \emptyset).$$

Then the steps in the program are (moving on one square after each instruction)

$$(S, \triangleright) \implies (I, \triangleright)[\text{move}](I, 0), \quad (I, 0) \implies (II, 0)[\text{move}](II, 1), \quad (II, 1) \implies (II, 0)[\text{move}](II, 1),$$
$$(II, 1) \implies (II, 0)[\text{move}](II, \emptyset), \quad (II, \emptyset) \implies (I, 1)[\text{move}](I, \emptyset), \quad (I, \emptyset) \implies (F, \emptyset)$$

At the end the tape will show

$$\triangleright : 0 : 0 : 0 : 1 : \emptyset : \emptyset : \cdots$$

From the two cases above (reading the tape data from the right), our program is executing the following mapping:

$$101 \to 111$$

$$110 \to 1000$$

so it's adding two (in binary two is 10) to the provided number, which is confirmed by executing the binary sums below:

$$101 + 10 = 111$$

$$110 + 10 = 1000$$

The Turing machine formalizes the idea of an algorithm: a set of procedures to evaluate a function.

The Church-Turing computability thesis indicates that "a function can only be calculated by effective means (on any classical computer) if it can be calculated on a Turing machine. Then, Turing machines allow a systematic study of the opration on all possible computers.

Remark 2. On a side note, the mathematician David Hilbert posed the following question (Entscheidungs-problem or decision-problem): "is there an algorithm with which we can determine whether or not any given mathematical statement is true?" The answer is that Turing showed there is no such function. He did it by asking if there is a function we can evaluate on a Turing machine, the value of which tells us whether or not any given program will terminate ("halting problem"). He showed that there isn't.

1.4. Vacuum tubes (valves)

They were devices useful as diodes, triodes, amplifiers, etc., used as switches in computers. They work by controlling the current flow between two electrodes (in high vacuum) over which there's an electric potential difference. They were invented by John Fleming (1904) and discontinued in 1960's. They produce a lot of heat and kept burning out. To give an example, the Eniac 1946 used 17,468 vacuum tubes.

1.5. Transistor

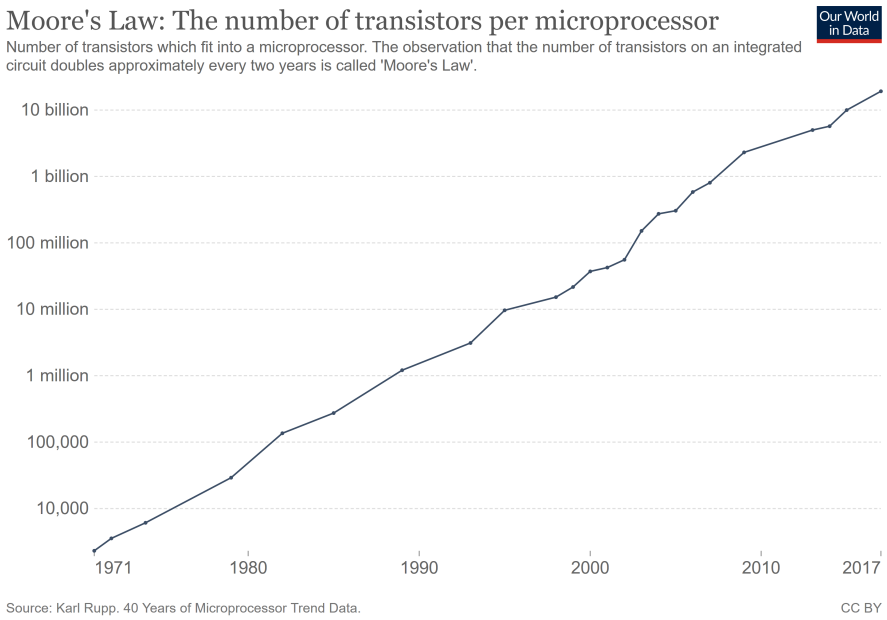
The first practical semiconductor transistor was invented by John Bardeen, Walter Brattain, and William Shockley in 1947, for which they were awarded the Nobel prize in Physics (1956).

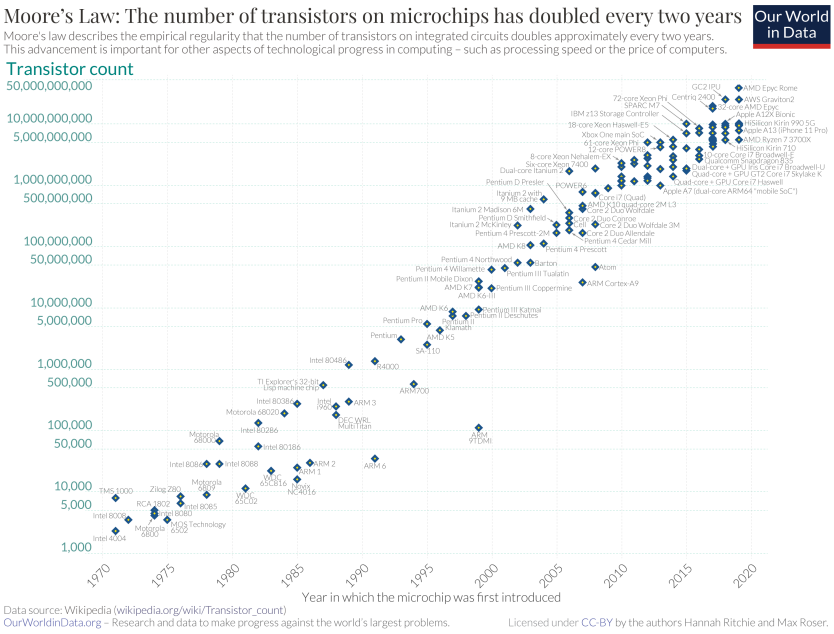
They're used as switches (on/off) and amplifiers. They were one of the greatest inventions of the 20th century as they revolutionized electronics. They're small, cheap to make, and reliable.

To give an example, a single logic gate used around 20 transistors, while an advanced microprocessor has about 3 billion of them. Many transistors are integrated on a single piece of silicon (integrated circuits).

1.5.1. Moore's Law

Proposed by Gordon Moore in 1964, he indicated by this law that the number of transistors that can be printed on a single circuit was increasing exponentially with time, doubling roughly every two years. For example, size features of around 5 nanometers were expected to be printable by 2021. The consequence of this size reduction law is that eventually the quantum scale on our classical computing devices is reached.





2. Introduction to Quantum computing

Quantum computing is fundamentally different from classical computing. One of the main takeaways from this course is to stress the point that information is PHYSICAL. We cannot consider it independently of the (physics of the) devices in which it's stored and the (physical) mechanisms with which it's modified. Information in quantum computers will need/use quantum physics. The main reason people is interested in quantum computers is that they are (theoretically / experimentally beginning to be) able to solve some problems (such as factorization of numbers) exponentially faster than classical computers.

2.1. Extended Church-Turing thesis

It indicates that the power of any computing device can only be polynomially faster that a regular "universal computer" (so this implies that any relative speedup can scale only according to a power law). Although performance of classical computers has increased by making operations faster (increased clock frequency), increasing the number of operations completed during each clock cycle.

However, Bernstein and Varizani (1993) proved that quantum computers could violate the extended Church-Turing thesis. In 1994, Peter Shor proposed an algorithm for a quantum computer that could factorize numbers exponentially faster than a classical computer (we'll study this later on in the course).

Remark 3. On a sidenote, Shor's algorithm was first executed in 2001, at IBM's Almaden Research Center & Stanford University. They factored $15{=}5{\times}3$ using 10^{18} identical molecules, each one containing seven active nuclear spins.

2.2. Some differences between classical and quantum computers

One of the first differences to point out is that classical (digital) computing uses bits as the fundamental building blocks for their computers. The state can be either 0 or 1, which is controlled by the voltage in that part of the circuit. They are robust against noise.

On the other hand, quantum computers use qubits, where actually the state can be 0 and 1 at the same time. Part of the experimental issues with quantum computers is that qubits are very sensitive to noise and decoherence.

2.3. Qubits

The basis states in quantum computing are $|0\rangle$ and $|1\rangle$. These are ket vectors in Quantum Mechanics (which uses the so-called "bra-ket" notation to represent functionals and vectors, respectively, of a Hilbert space over the complex numbers). Physically, these basis states could concretely be either two energy levels of an atom, two polarization states of a photon, or two spin states of an electron.

As you might know, quantum mechanics is modelled by a linear PDE called the Schroedinger equation (it looks like a diffusion equation with a complex coefficient), and therefore it allows to have a Superposition Principle, meaning that

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

is in principle an allowed state, where the coefficients α and β represent probability "amplitudes" (notice the last word as it's important). We have to stress that the wave-function $\Psi(x,t) = \langle x\,|\Psi\rangle\,(t)$ is complex-valued, so the probability amplitudes can be complex numbers (with a norm and a phase, as for a complex number $z = r\exp(i\theta)$). Therefore interference, as in any wave-phenomena, is possible.

Now, Quantum Mechanics uses normalized state-vectors, meaning that the normalization condition

$$\langle\Psi\,|\Psi\rangle = 1$$

must hold. This implies (under the knowledge that our basis states are orthonormal under the inner/vector product chosen) that

$$1 = \langle\Psi\,|\Psi\rangle = \langle\alpha\,|0\rangle + \beta\,|1\rangle\;|\alpha\,|0\rangle + \beta\,|1\rangle\rangle = \alpha^*\alpha\,\langle 0\,|0\rangle + \alpha^*\beta\,\langle 0\,|1\rangle + \beta^*\alpha\,\langle 1\,|0\rangle + \beta^*\beta\,\langle 1\,|1\rangle = |\alpha|^2 + |\beta|^2$$

This is why α and β are called probability amplitudes: the squares of their norm represent probabilities, the sum of them adding to one. As a consequence, we notice that

$$0 \le |\alpha|^2 \le 1, \quad 0 \le |\beta|^2 \le 1,$$

as for any set of probabilities.

2.3.1. Bloch sphere

We will represent our state vector

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

in the particular form (let's assume that α is real-valued, without actually losing generality because the wavefunction is defined up to a phase anyways)

$$|\Psi\rangle = \cos(\frac{\theta}{2})\,|0\rangle + e^{i\phi}\sin(\frac{\theta}{2})\,|1\rangle$$

where we have made the change of variables (α is taken real but β can be complex which we represent in polar form)

$$\alpha = \cos(\frac{\theta}{2}) \iff \theta = 2\arccos(\alpha)$$

$$\beta = e^{i\phi}\sin(\frac{\theta}{2}) \iff \phi = \ln(\frac{\beta}{\sin(\arccos(\alpha))})/i$$

Notice that the choices above satisfy

$$|\alpha|^2 + |\beta|^2 = \cos^2(\theta/2) + \sin^2(\theta/2) = 1.$$

Any state-vector obtained by the superposition of our 2D basis can be graphically represented via these angular variables, as one can see below, as a point in the so-called "Bloch sphere":

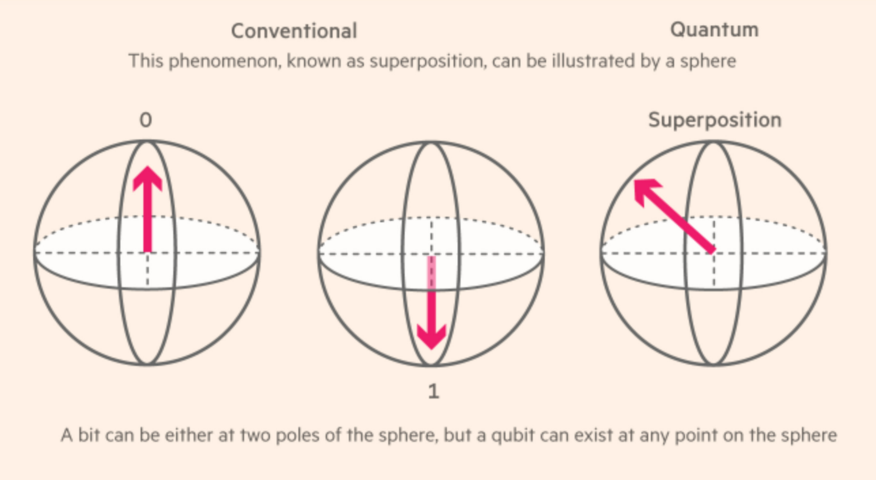


Fig. From "Overview of quantum computing: progress and prospects", by Prof. Duncan O'Dell, Introduction to Quantum Information course, McMaster University, 2019.

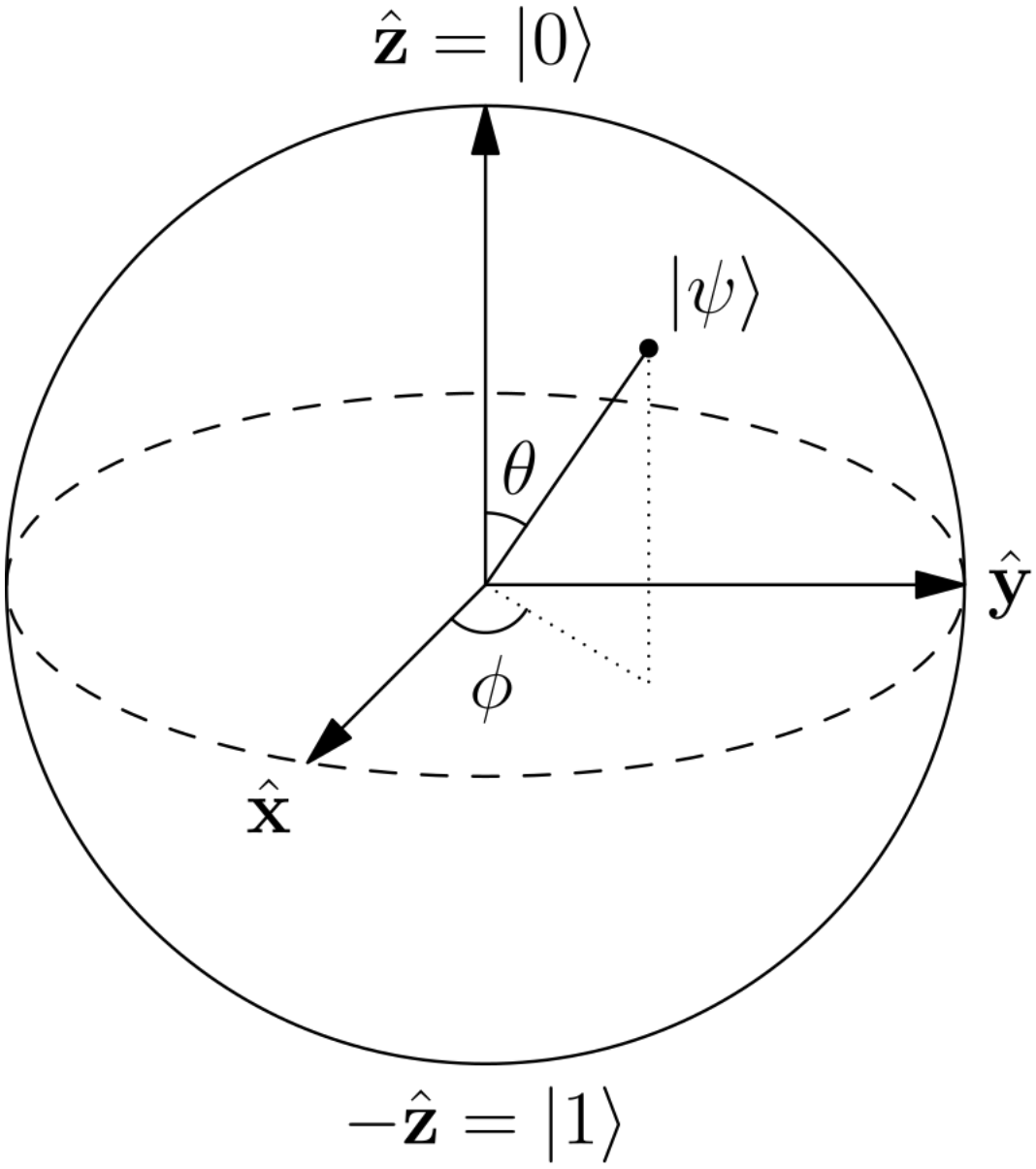


Fig. By Glosser.ca - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=23263326

Notice the representations of the basis vectors in this sphere (lying on its poles), and an example of their superposition above (a point on the sphere of norm 1 since $\langle \Psi | \Psi \rangle = 1$).

The "digital" properties of qubits rely on the basis states (as for bits), while the "analog" properties rely on the amplitudes. The fact that qubits have both properties makes them sensitive to noise/decoherence.

2.3.2. Requirements for a computer

We would require

- Memory
- Initialization (prepare register in the desired state, for example, all at zero)
- Capability to perform universal logical operations
- VERY Small error rate that can be fixed by "error correction" methods
- Capability to read out the final result.

2.3.3. Quantum measurements

In the case of classical bits, we can measure the state of a bit, getting 0 or 1, with the measurement process not changing the state of this bit.

However, in quantum computing & information, we cannot measure observables without perturbing (this is one of the keypoints of quantum mechanics). That means that for

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

the measurement will change α, β. In fact, given the state (assuming we measure the observable for which $|0\rangle$, $|1\rangle$ are the basis states associated to its operator), after the measurement we will get $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$. In summary, we know from quantum mechanics that the measurement will change the state of our qubit.

2.3.4. Multiple qubits

What happens when we have two (qu-)bits? We will have four basis states, namely

$$|00\rangle\,, \quad |01\rangle\,, \quad |10\rangle\,, \quad |11\rangle$$

As we know quantum mechanics allows superposition, so the state vector representing the two (qu-)bits can be in a superposition of the $4 = 2^2$ possible states in the basis above:

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

Let's generalize this to the case of N (qu-)bits. We have N "slots" with 2 possible choices of bits for each one (either 0 or 1), so the number of possible (basis) states is $\underbrace{2 \times \cdots \times 2}_{\text{N times}} = 2^N$, since

$$\left|\underbrace{- \cdots -}_{\text{N slots}}\right\rangle$$

We have done before the cases $N = 1, 2$. The origin of "quantum speed up" is the related massively parallel computing (we will explain this more in detail during the course). To give an idea of the exponential behavior of the number of states, think about this:

1. With 2 qubits, we have $2^2 = 4$ amplitudes.
2. With 3 qubits, we have $2^3 = 8$ amplitudes.
3. With 10 qubits, we have $2^{10} = 1024$ amplitudes.
4. With 20 qubits, we have $2^{20} = 1048576$ amplitudes.
5. With 30 qubits, we have $2^{10} = 1073741824$ amplitudes.
6. With 500 qubits, we have more amplitudes than the estimated number of atoms in the Universe!

2.4. Entanglement: an introduction

Product states are of the form

$$|\psi_1\rangle \otimes |\psi_2\rangle$$

Loosely speaking the intuition about product states is that there's some independence of probabilities between $|\psi_1\rangle\,, \quad |\psi_2\rangle$. An example is (it's the same example just written in different notations):

$$|0\rangle \otimes |0\rangle = |0\rangle_a \otimes |0\rangle_b = |0\rangle\,|0\rangle = |00\rangle$$

On the other hand, an entangled state is by definition a state that cannot be expressed as a product state for any $\psi_1, \quad \psi_2$.

Consider the case where we have two qubits (with 4 basis states). The linear combination below is an example of an entangled state

$$|\Psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq |\psi_1\rangle \otimes |\psi_2\rangle$$

Notice the state above is normalized (under orthonormality):

$$\langle\Psi\,|\Psi\rangle = \frac{\langle 00\,|00\rangle + \langle 11\,|11\rangle}{\sqrt{2}^2} = \frac{2}{2} = 1$$

The state above has a probability $(1/\sqrt{2})^2 = 1/2$ of being in the basis state $|00\rangle$ and $|11\rangle$ respectively at the same time. The interesting thing about entanglement, as we shall see later on in the course, is that although our two qubits might be spatially separated, if they are in the entangled state above, measurement on one of them immediately collapses the state of the other. This is a phenomena that Einstein called "Spooky action at distance". This correlation is one of the most interesting features of quantum mechanics. Measurement results will still be random but correlated, as we will see when we study Bell's inequality.

2.5. Quantum computing

- A quantum computer operates at the same time on many different numbers, but in a different way than a classical computer doing processes in parallel.
- Each time a quantum computer is run, one can get different answers even when starting from the same initial state...
- Key to quantum algorithms:
  - use (destructive) interference to remove undesired numbers
  - use (constructive) interference to increase the probability of desirable numbers
  - due to entanglement, measurements of different qubits are highly correlated
- Example: Shor's factoring algorithm. It doesn't always give the correct answer exactly, but the algorithm increases the probability of getting the right one. On the other hand, the reverse process of checking if the answer is right (as for $3 \times 5 = 13$) is quick to do.

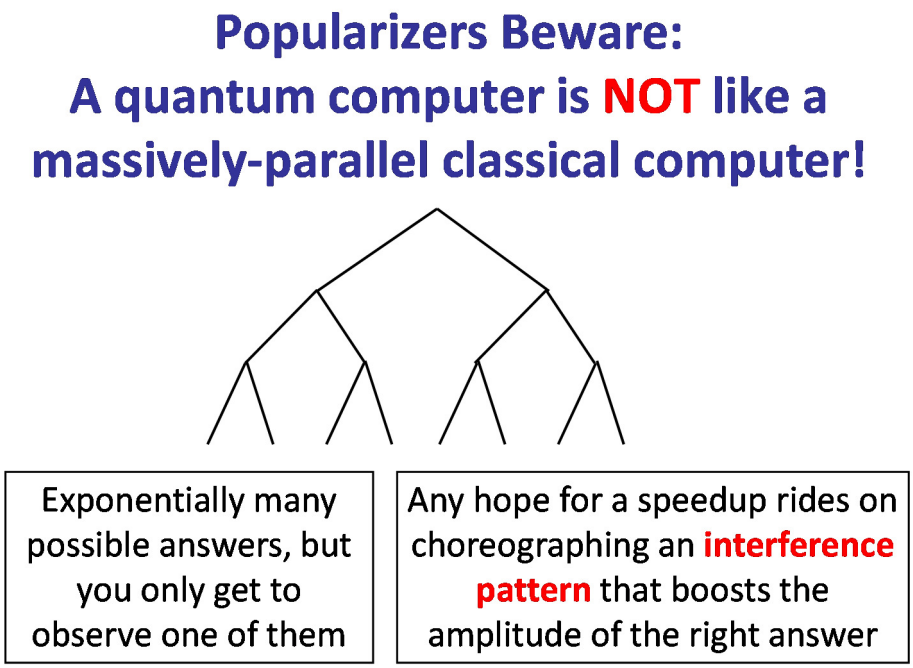


Fig. from talk of Prof. Scott Aaronson (University of Texas at Austin) at UT San Antonio, Spring 2021, "Quantum Computational Supremacy".

2.6. Approach to quantum computing

The main approach to quantum computing is based on logic gates. It is adapted from classical computing, and has been experimentally realized, for example with laser-cooled ions.

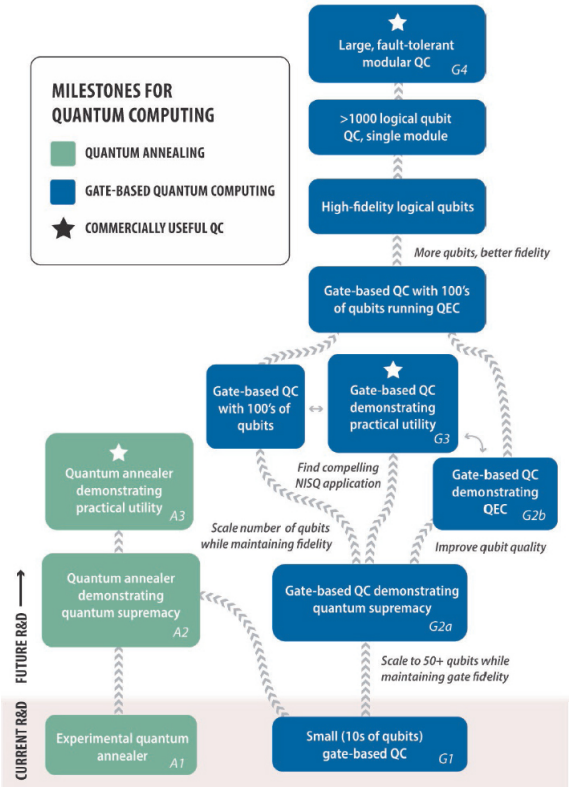
There's an alternate approach, called analogue quantum computing, that includes quantum annealing and adiabatic quantum computing. It's used by companies like D-wave, whose approach was initially popular, but current opinion now is that its approach has been surpassed by others. The idea in general is to change the Hamiltonian with time slowly (adiabatically), changing the state slowly to the right answer.

2.7. Experimental work on quantum computers

- Laser-cooled trapped ions:
  - The first two-qubit logic gate was made with trapped ions at NIST in Colorado. Look at C. Monroe, D.M. Meekhof, B.E. King, W.M. Itano & D.J. Wineland, Phys. Rev. Lett. 75, 4714 (1995).
  - Algorithms demonstrated with this approach are, for example, Shor's factoring, Quantum Fourier transform, and Grover's search.
- Superconducting qubits (has been used by Google, Intel, D-wave machine)
- Photons (where the entanglement is generated through optical nonlinearity or by measurement)
- Topological qubit (based on topological superconductors/insulators with Majorana fermions): this approach avoids decoherence

2.8. Milestones for Quantum Computing

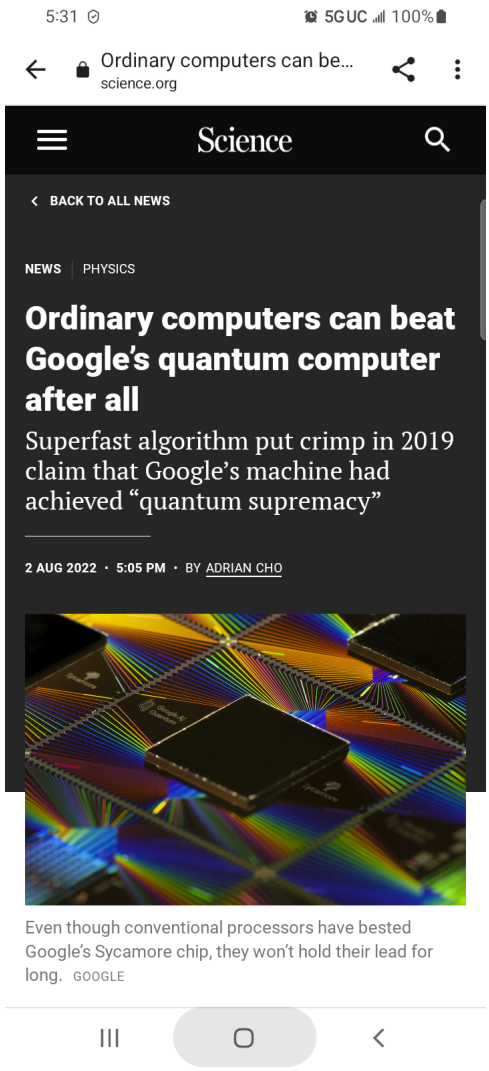
Look at the following figure from the report "National Academies of Sciences, Engineering, and Medicine. 2019. Quantum Computing: Progress and Prospects. Washington, DC: The National Academies Press. https://doi.org/10.17226/25196."



2.9. Quantum Computing & Information currently...

It must be said that Quantum Computing and Information is a very hot topic nowadays, with many international efforts on many fronts and going at a very fast pace.

It's growing so fast that one must check weekly the ArXiv to catch up with a current literature review...



References

[1] "Overview of quantum computing: progress and prospects", by Prof. Duncan O'Dell, Introduction to Quantum Information course, McMaster University, 2019.

[2] Quantum Information, by Stephen M. Barnett (Oxford University Press, ISBN 978-0-19-852763-3).

[3] https://ourworldindata.org

[4] Karl Rupp. 40 Years of Microprocessor Trend Data. Retrieved 2017-11-30. Link https://www.karlrupp.net/2015/06/40-years-of-microprocessor-trend-data/

[5] "Quantum Computational Supremacy", talk by Scott Aaronson (University of Texas at Austin) at UT San Antonio, Spring 2021.

[6] National Academies of Sciences, Engineering, and Medicine. 2019. Quantum Computing: Progress and Prospects. The National Academies Press, Washington, DC. doi: https://doi.org/10.17226/25196.

[7] By Glosser.ca - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=23263326