Welcome to **MULT90063**

“Introduction to Quantum Computing”

University of Melbourne

Subject summary 2021

24 Lectures:

Tuesday 9am (start 9.05am): Redmond Barry-102 (Latham Theatre) (dual delivery)
Wednesday 9am (start 9.05am): Physics South-L108 (Laby Theatre) (dual delivery)

12 Computer-based lab classes (commencing in week 1):

Friday, Baker Computer Lab, Level 6, David Caro Building or online (dual delivery)

Assessment (see Subject Outline on LMS for timing):

2 assignments (30% and 30%) , two-hour exam (40%)

Lecturers/coordinators: Dr. Charles Hill (cdhill@unimelb.edu.au)
Dr. Casey Myers
Dr. Muhammad Usman
Prof. Lloyd Hollenberg

LMS: Access lecture notes, lab notes, assignments (recommend bringing print-out of lab notes to labs)

Suggested reading:

E. Rieffel and W. Polak – "Quantum Computing: A Gentle Introduction"

P. Kaye, R. Laflamme and M. Mosca – "An Introduction to Quantum Computing"

M. Nielsen & I. Chuang – "Quantum Computation and Quantum Information"

Opportunities for interaction

- Ask questions during lectures
- Ask questions after lectures
- Ask your lab class demonstrators
- Ask on the discussion board on LMS

Assessment

Description	Timing	Percentage
Project 1	Week 6	30%
Project 2	Week 12	30%
Examination (2 hour)	During the examination period	40%

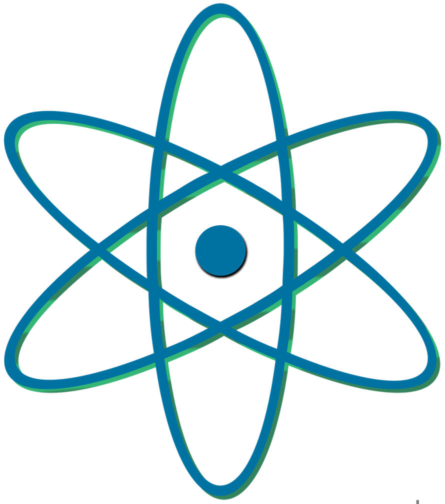
Week by week

(1) Introduction to quantum computing
(2) Single qubit representation and operations
(3) Two and more qubits
(4) Simple quantum algorithms
(5) Quantum search (Grover's algorithm)
(6) Quantum factorization (Shor's algorithm)
(7) Quantum supremacy and noise
(8) Programming real quantum computers (IBM Q)
(9) Quantum error correction (QEC)
(10) QUBO problems and Adiabatic Quantum Computation (AQC)
(11) Variational/hybrid quantum algorithms (QAOA and VQE)
(12) Solving linear equations, QC computing hardware

Classical mechanics and Quantum mechanics

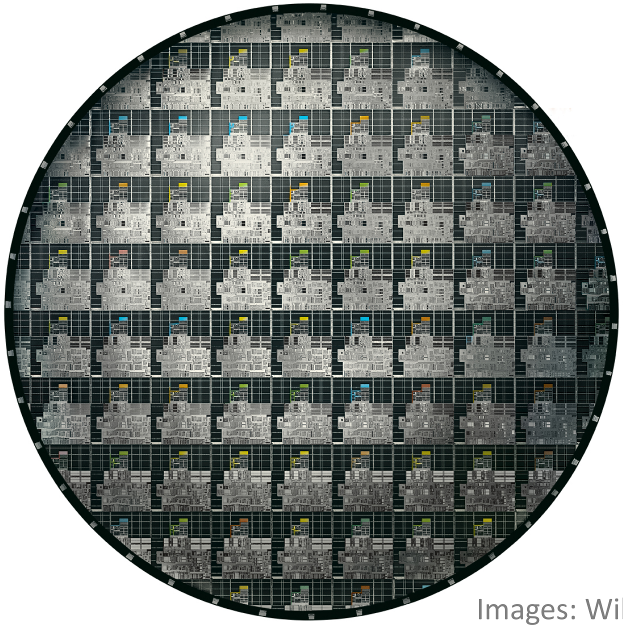
“Classical” mechanics rules our medium scale world.

“Quantum” mechanics rules the world of atoms and molecules.



Images: Pixabay

Information is physical



Images: Wikimedia commons

Your computer is a *physical* device.

It operates according to the laws of ***physics***.

Moore's Law

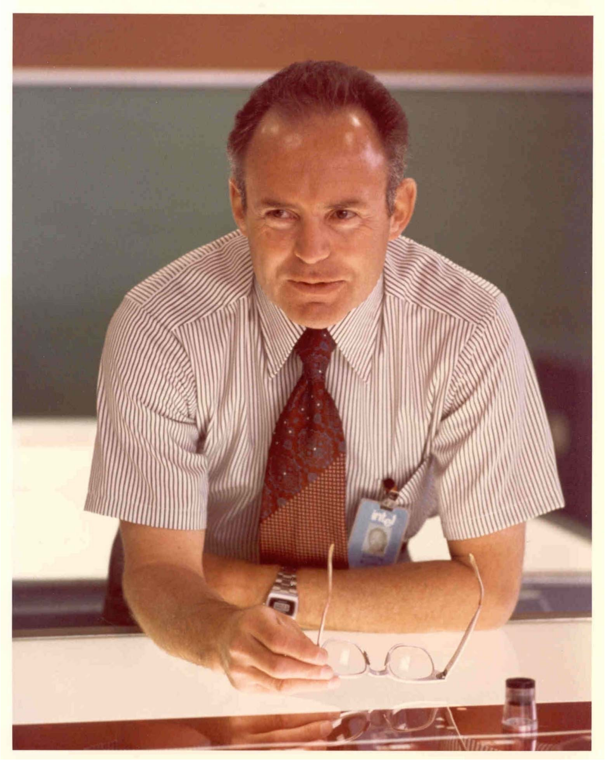


Image: Wikimedia commons

"The number of transistors incorporated in a chip will approximately double every 24 months."

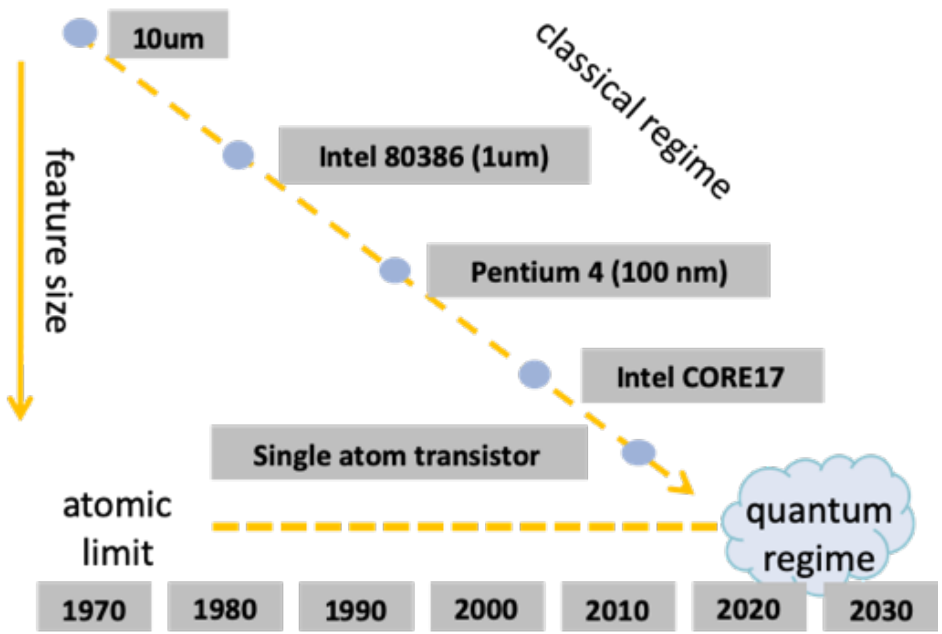
— Gordon Moore, 1965

The limits of conventional computing

Conventional computers: some problems are just too hard to solve on even the biggest supercomputers -> More powerful CPUs?

Moore's Law – since 1960's transistor size halves every 1.5 years

As transistor size < 10nm we're getting close to the limit (atoms are about 1nm or less).



The end-of-Moore's law signals new computer technology is needed…

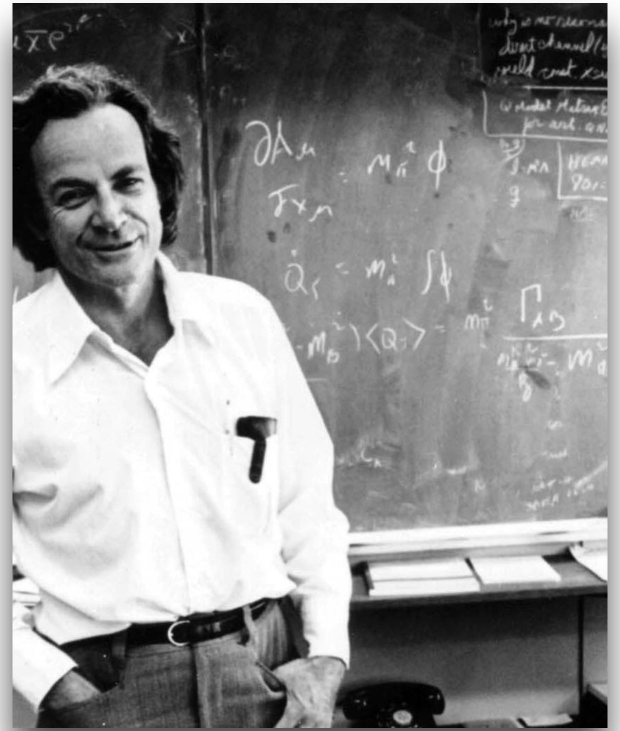


Image: Wikimedia commons

I am not afraid to consider the final question as to whether, ultimately – in the great future – we can arrange the atoms the way we want; the very atoms, all the way down!

There's plenty of room at the bottom — Richard Feynman

Quantum Mechanics

Classical mechanics

- Newton's laws, currents, 'classical' computing….

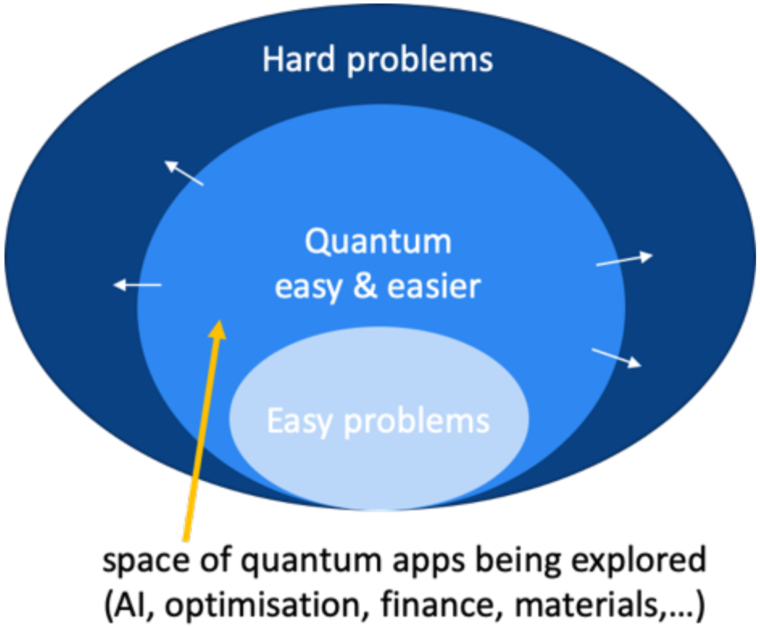
Quantum mechanics

- Entanglement, tuneling, Quantum computing,…

Smaller

Quantum Computing gives an advantage!

Surprisingly, for some problems quantum mechanics allows **more efficient** solutions than are possible using classical mechanics.



Quantum computers based on the laws of quantum mechanics circumvent limitations of classical information processing

MULT90063: Introduction to Quantum Computing

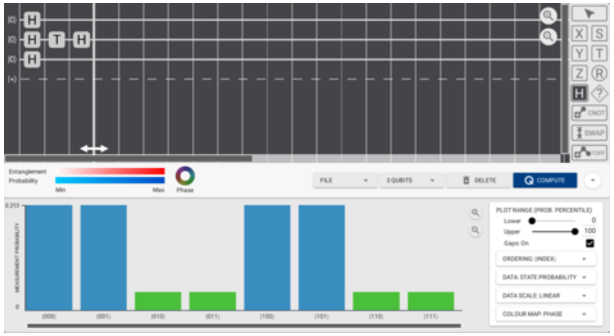
This subject is a introduction to quantum computing.

In this subject we will cover:

- The basic concepts of quantum information and quantum logic
- Programming quantum computers to perform computational tasks
- Quantum algorithms
- Quantum error correction
- The status of current quantum computer technology

Teaching system: UoM Quantum User Interface (QUI)

- Prior to Lab Class 1: access and sign in at *quispace.org* *(using your unimelb email address)*
- In Lab Class: registered users have expanded capabilities

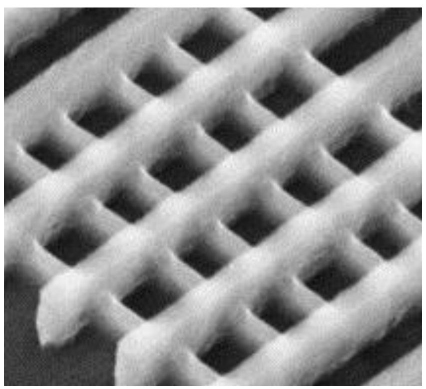


Bits and qubits

"Classical" computing: Bits

Conventional computers obey the laws of "classical physics" (pre-quantum).

A classical **bit** (e.g. in an Intel chip) is like a switch with two values:



Intel.com

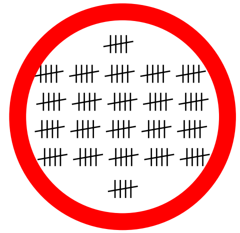
on "1"

off "0"

Only "0" <u>or</u> "1" at a time in any given physical representation (transistor)

From classical bits → we can construct binary numbers eg. 0000 <u>or</u> 1010

Decimal notation



c. 100,000 BC

Roman

Our number system is decimal – based on powers of 10.
Computers work on a binary number system – base 2.

We know instinctively that "**110**" is shorthand for:

1x100 + **1**x10 + **0**x1 = "one hundred and one ten and no ones"

If we remember we're in base 10, we write it as:

1x10^2 + **1**x10^1 + **0**x10^0

= one of 10^2 plus one of 10^1 plus zero of 10^0

$10^2 = 100$
$10^1 = 10$
$10^0 = 1$

"one hundred and ten"

"six"

But, in base 2 the number "110" is actually "six", i.e.

Decimal and binary notation

THE UNIVERSITY OF MELBOURNE

Decimal

$10^2 = 100$
$10^1 = 10$
$10^0 = 1$

110
base 10

"one hundred and ten"

$110 = 1 \times \underline{10}^2 + 1 \times \underline{10}^1 + 0 \times \underline{10}^0$ = one of $\underline{10}^2$ plus one of $\underline{10}^1$ plus zero of $\underline{10}^0$
$= 100 + 10 + 0 = 110$

Binary

$2^2 = 4$
$2^1 = 2$
$2^0 = 1$

110
base 2

"six"

$110 = 1 \times \underline{2}^2 + 1 \times \underline{2}^1 + 0 \times \underline{2}^0$ = one of $\underline{2}^2$ plus one of $\underline{2}^1$ plus zero of $\underline{2}^0$
$= 4 + 2 + 0 = 6$

A number, n, with digits, d_k, and base, b has a value of

$$n = \sum_k d_k b^k$$

There are 10 types of people in this world…
– those who understand binary numbers…and those who don't

Some examples

Base 10
"decimal"

57

57 = 5x10^1 + 7x10^0 = five of 10^1 plus seven of 10^0 = 50 + 7 = 57

Note: the numbers in blue run over base 10 "digits" i.e. 0-9

57
1's
10's

$10^3 = 1000$
$10^2 = 100$
$10^1 = 10$
$10^0 = 1$

Base 2
"binary"

111001

111001 = 1x2^5 + 1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0

= 1x32 + 1x16 + 1x8 + 0x4 + 0x2 + 1x1

= 32 + 16 + 8 + 0 + 0 + 1 = 57

111001
1's
2's
4's
8's
16's
32's

$2^5 = 32$
$2^4 = 16$
$2^3 = 8$
$2^2 = 4$
$2^1 = 2$
$2^0 = 1$

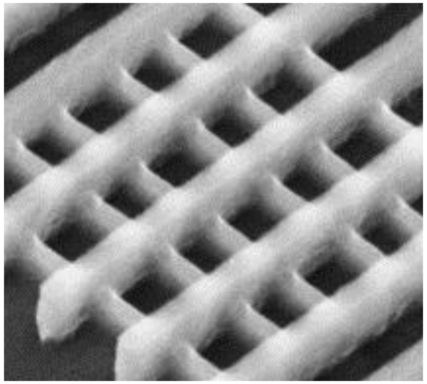
Note: the numbers in blue run over base 2 "bits" i.e. 0-1

The point: as computers naturally use binary numbers, so do quantum computers

→ need to be familiar with bits and binary and decimals (see table)

Decimal	Binary	Decimal	Binary
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

“Classical” computing: bits and binary numbers



Intel.com

Only “0” <u>or</u> “1” at a time in any given physical representation (transistor)

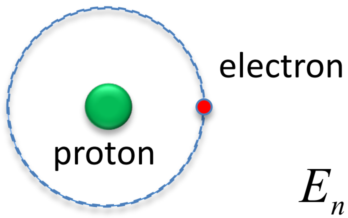
From classical bits → we can construct binary numbers 0000 <u>or</u> 1010 etc

One binary number at one time in the physical representation

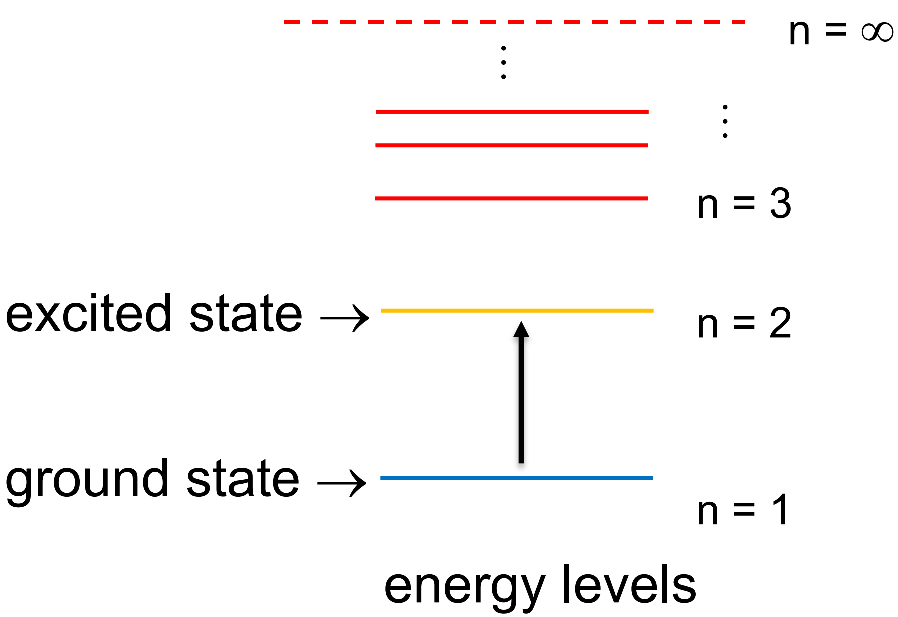
Start with a (semi) familiar concept – the atom

Planck (1900): postulated that energy is quantised

Bohr (1913): constructed a simple "quantised" model of the hydrogen atom



$$E_n = -\frac{m}{2\hbar^2}\left(\frac{e^2}{4\pi\,\varepsilon_0}\right)^2\frac{1}{n^2}$$

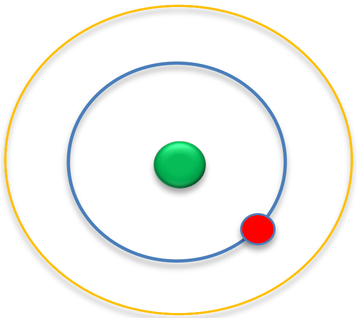


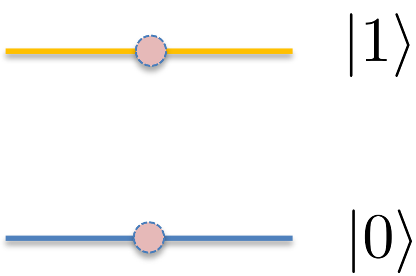
Consider lowest two levels

Simply re-label states to bit notation

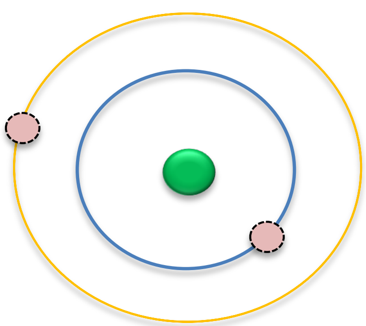
A quantum bit, or "**qubit**" is a two level quantum system

Quantum superposition and measurement

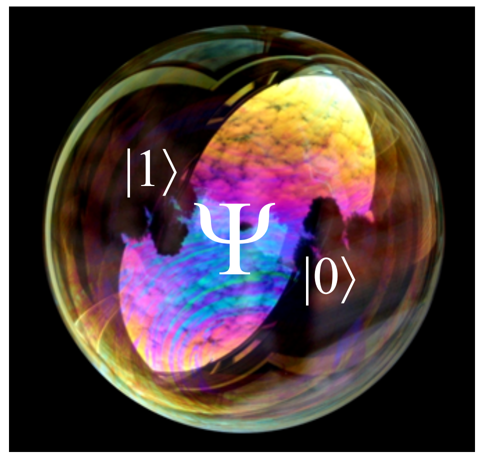




One electron in quantum superposition.

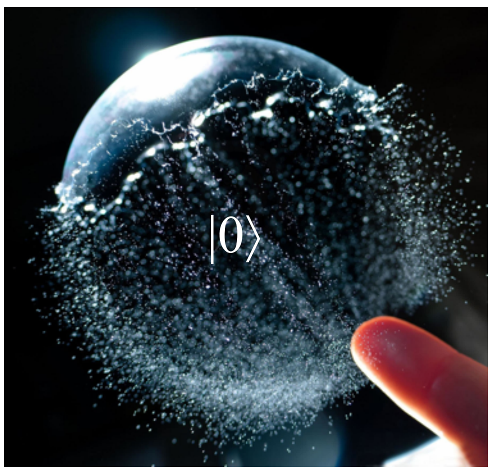


quantum superposition



$|0\rangle$ “and” $|1\rangle$

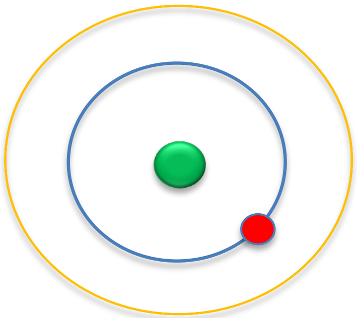
measurement/observation

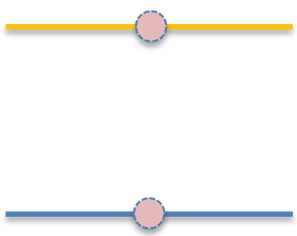


state collapse $\rightarrow$ random outcome

Bubble: Brocken commons.wikimedia.org
Bursting bubble: wallpaperswide.com

Quantum superposition and measurement

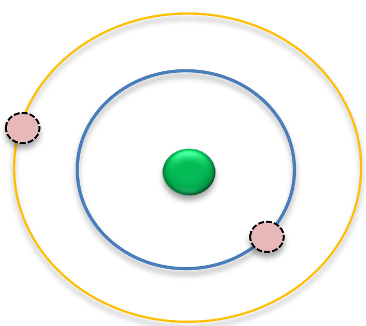




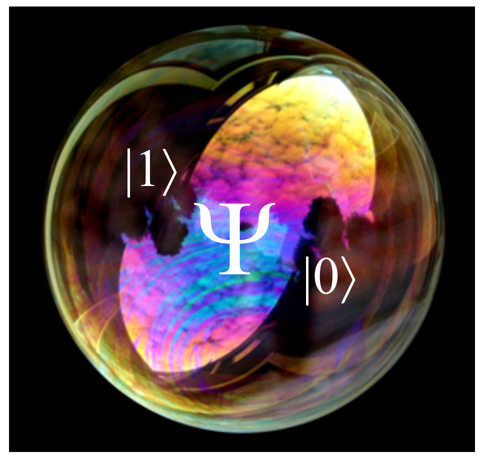
$|1\rangle$

$|0\rangle$

One electron in quantum superposition.



quantum superposition



$|0\rangle$ "and" $|1\rangle$

measurement/observation

state collapse $\rightarrow$ random outcome

Bubble: Brocken commons.wikimedia.org
Bursting bubble: wallpaperswide.com

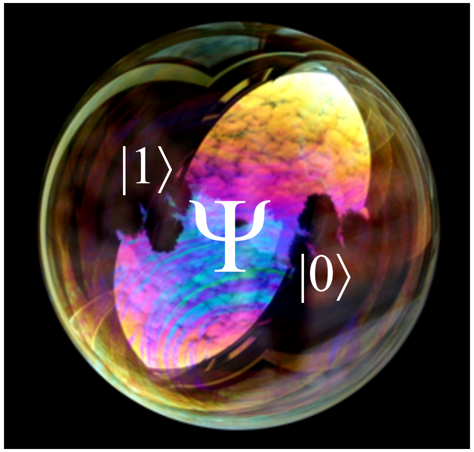
Quantum Bits = Qubits

Instead of bits we have **qubits.**

As we will see, qubits can be in **superpositions:**

$$\alpha|0\rangle + \beta|1\rangle$$

where α and β are called **amplitudes,** and are complex numbers.



If we were to measure we would randomly measure “0” with probability:

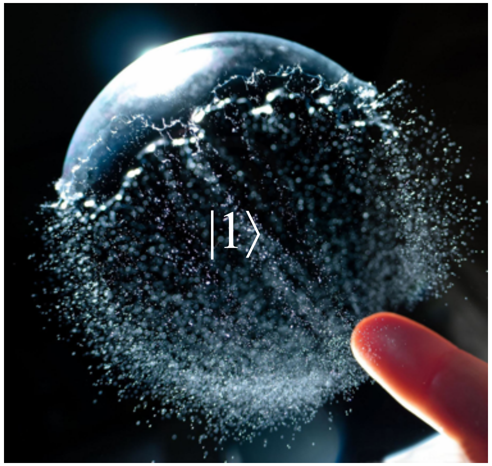
$$|\alpha|^2$$

or “1” with probability,

$$|\beta|^2$$

Since probabilities must sum to 1,

$$|\alpha|^2 + |\beta|^2 = 1$$

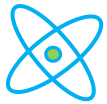


measurement/observation state collapse → random outcome

Combining qubits – binary notation

Binary combinations

1 qubit:

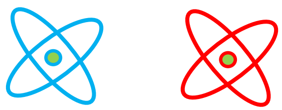


$|0\rangle, |1\rangle$

$|0\rangle, |1\rangle$

binary representation of decimals 0 to 1

2 qubits:

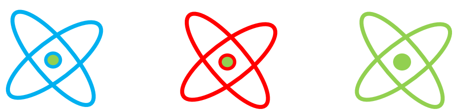


$|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$

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

binary representation of decimals 0 to 3

3 qubits:

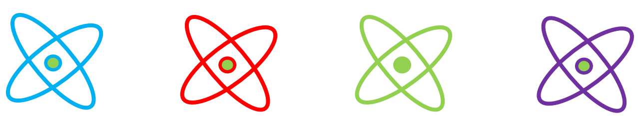


$|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$

$|000\rangle, |001\rangle, |010\rangle, |011\rangle$
$|100\rangle, |101\rangle, |110\rangle, |111\rangle$

binary representation of decimals 0 to 7

4 qubits:



$|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$ $\quad$ $|0\rangle, |1\rangle$

$|0000\rangle, |0001\rangle, |0010\rangle, |0011\rangle$
$|0100\rangle, |0101\rangle, |0110\rangle, |0111\rangle$
$|1000\rangle, |1001\rangle, |1010\rangle, |1011\rangle$
$|1100\rangle, |1101\rangle, |1110\rangle, |1111\rangle$

binary representation of decimals 0 to 15

Each of these states has an associated amplitude.

A general quantum superposition

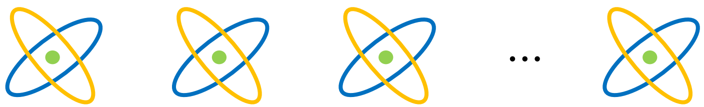
In a quantum computer, we can be in a superposition of many different states. The state of a quantum computer is a superposition over all binary strings, “k”,

$$|\psi\rangle = \sum_k a_k \, |k\rangle$$

amplitudes — states (binary string)

Multiple qubits and quantum processing

Basic <u>representation</u> of binaries as quantum information:

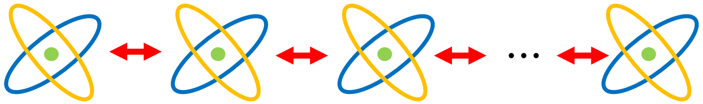


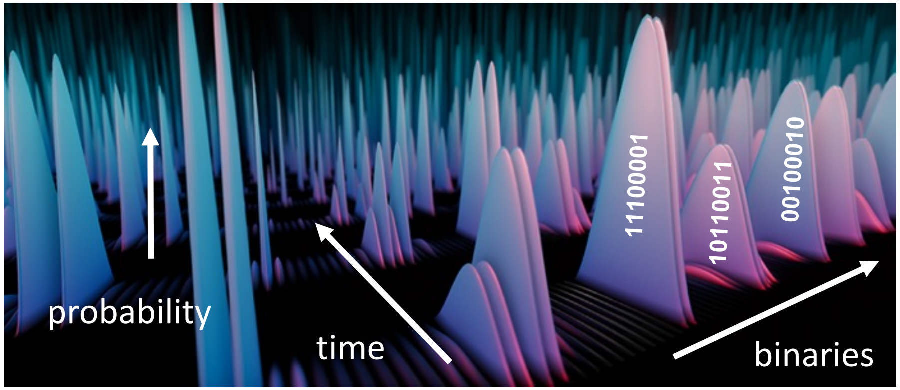
N qubits

Independent quantum superpositions $\rightarrow$ superposition over N-bit binaries $|000\ldots0\rangle,\ldots, |111\ldots1\rangle$ (and there are 2^N of these)

Not very useful…measurement of qubits collapses to one random N-bit string

<u>Quantum computation</u>: qubits interact to create complex superpositions and entangled states



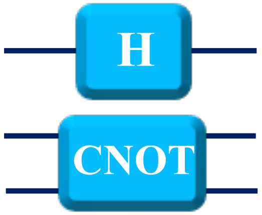


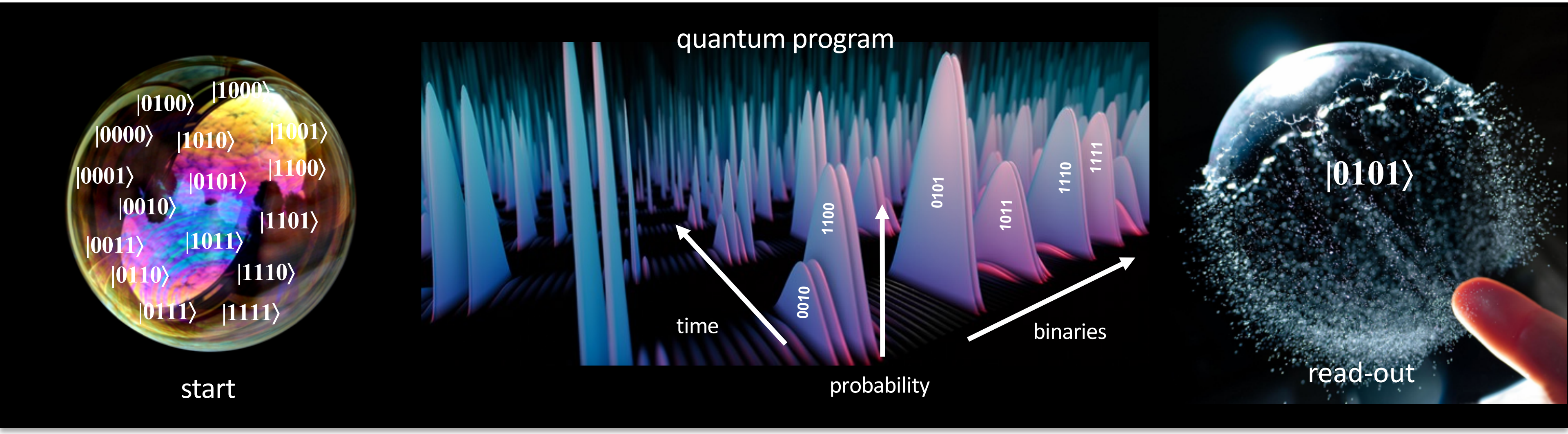
Bubble: Brocken commons.wikimedia.org

Quantum information processing

Digital quantum computing:

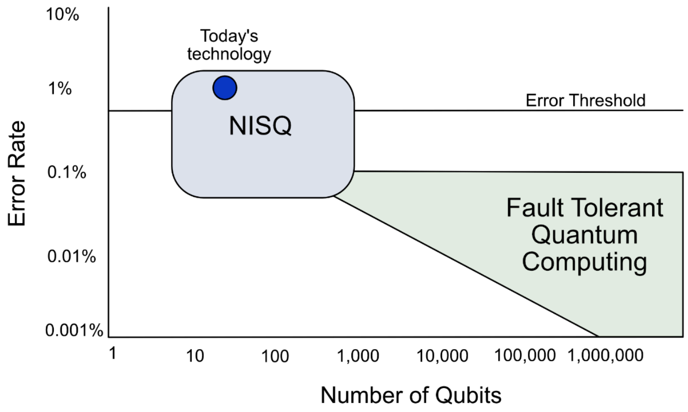
- logic gates between qubits perform mathematical operations on binary data
- complex states created → binary data in a large superposition of many states
- quantum interference amplifies probability of desired output (answer)

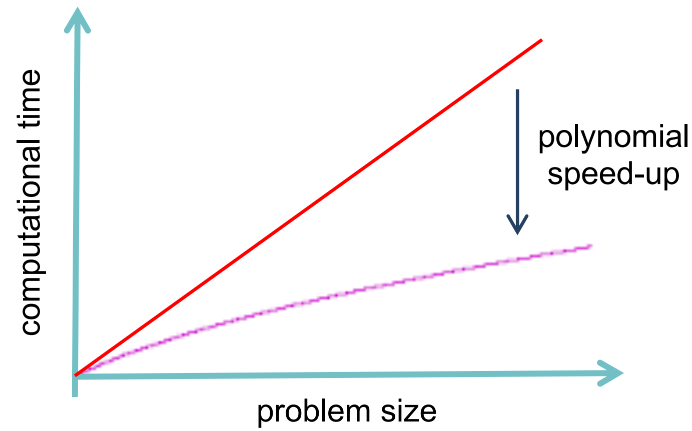




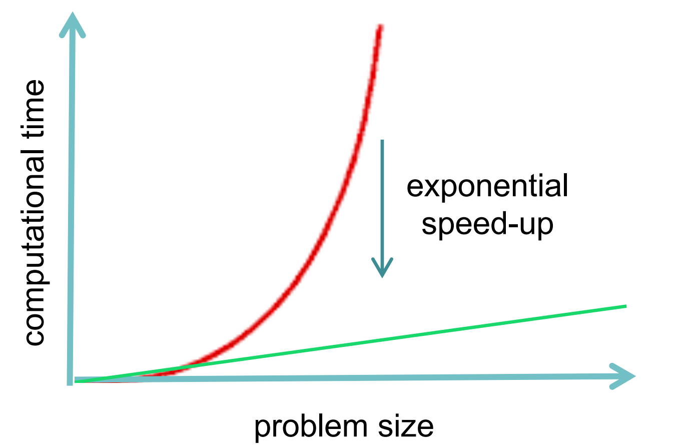
Bubble: Brocken commons.wikimedia.org
Bursting bubble: wallpaperswide.com

NISQ Devices





- **Noisy Intermediate Scale Quantum (NISQ)**
  - medium-scale system (10 ~1000 qubits)
  - simplified control and error correction
  - potentially polynomial speed-up (e.g. √CPU)
  - optimisation, machine learning, chemistry,…
  - pathway to full-scale universal QC…



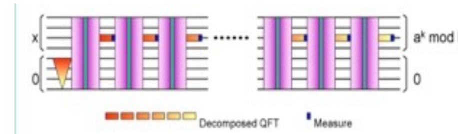
- **Fault Tolerant Quantum Computing**
  - large-scale system (>>1000 qubits)
  - high redundancy for quantum error correction
  - potentially exponential speed-up for some problems (e.g. Shor's factoring algorithm)

Timeline and state of the art

Google "Quantum Supremacy", random quantum circuit: 53 qubits

Quantum advantage 100-1000 qubits?

increasing # qubits and quality (error rates)



Universal QC: (millions of qubits) poly to exponential speed-up on various problems

Hardware race: IBM, Google, Intel, D-Wave, Rigetti, Microsoft, SQC,…

DWave: 2000 "qubits" "analogue QC"

IBM-Q: 65 qubits digital QC

In the meantime, the era of quantum software and app development has begun…

Week by week

(1) Introduction to quantum computing
(2) Single qubit operations and representation
(3) Two and more qubits
(4) Simple quantum algorithms
(5) Quantum search (Grover's algorithm)
(6) Quantum factorization (Shor's algorithm)
(7) Quantum supremacy and noise
(8) Programming real quantum computers (IBM Q)
(9) Quantum error correction (QEC)
(10) QUBO problems and Adiabatic Quantum Computation (AQC)
(11) Variational/hybrid quantum algorithms (QAOA and VQE)
(12) Solving linear equations, QC computing hardware