# From Billiard Balls to Quantum Computing:

## a tutorial on the foundations of computing

Geoff Sharman

# My qualifications for giving this talk?

- Ph.D. in Particle Physics
- 35 years in IBM research & development
- Lots of reading!
- But I'm not an expert on QC ...

# Dramatis Personae

- Alan Turing, Cambridge university
- Rolf Landauer, IBM Research Yorktown NY
- Charles Bennett, IBM Research Yorktown NY
- Richard Feynman, Caltech
- David Deutsch, Oxford University

# Alan M Turing

- *On computable numbers, with an application to the Entscheidungsproblem* [decision problem] (1936)
- Showed that computing is a physical process [so subject to $2^{nd}$ Law of Thermodynamics]
- Showed that computing machines are universal, i.e. can simulate any machine in a finite number of steps, including any other computer

# Rolf Landauer

- *Irreversibility and Heat Generation in the Computing Process* (1961)

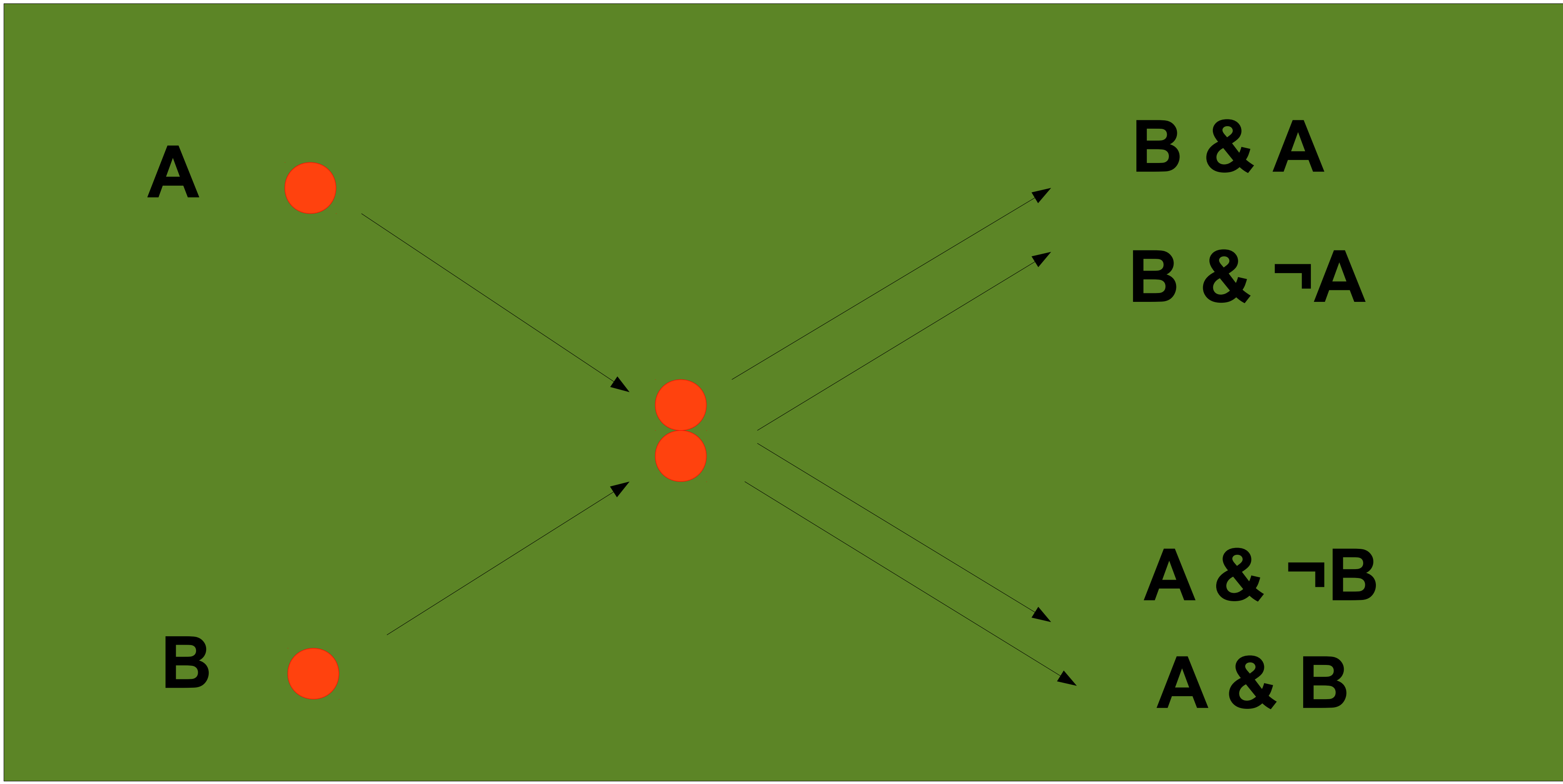
- Wanted to understand the minimum amount of energy required per computational step
  - showed that at least kT log2 energy is expended when 1 bit is discarded (known as the Landauer limit)
  - where k is Boltzmann's constant and T is temperature

- Showed that “information is inevitably physical”

# Charles Bennett

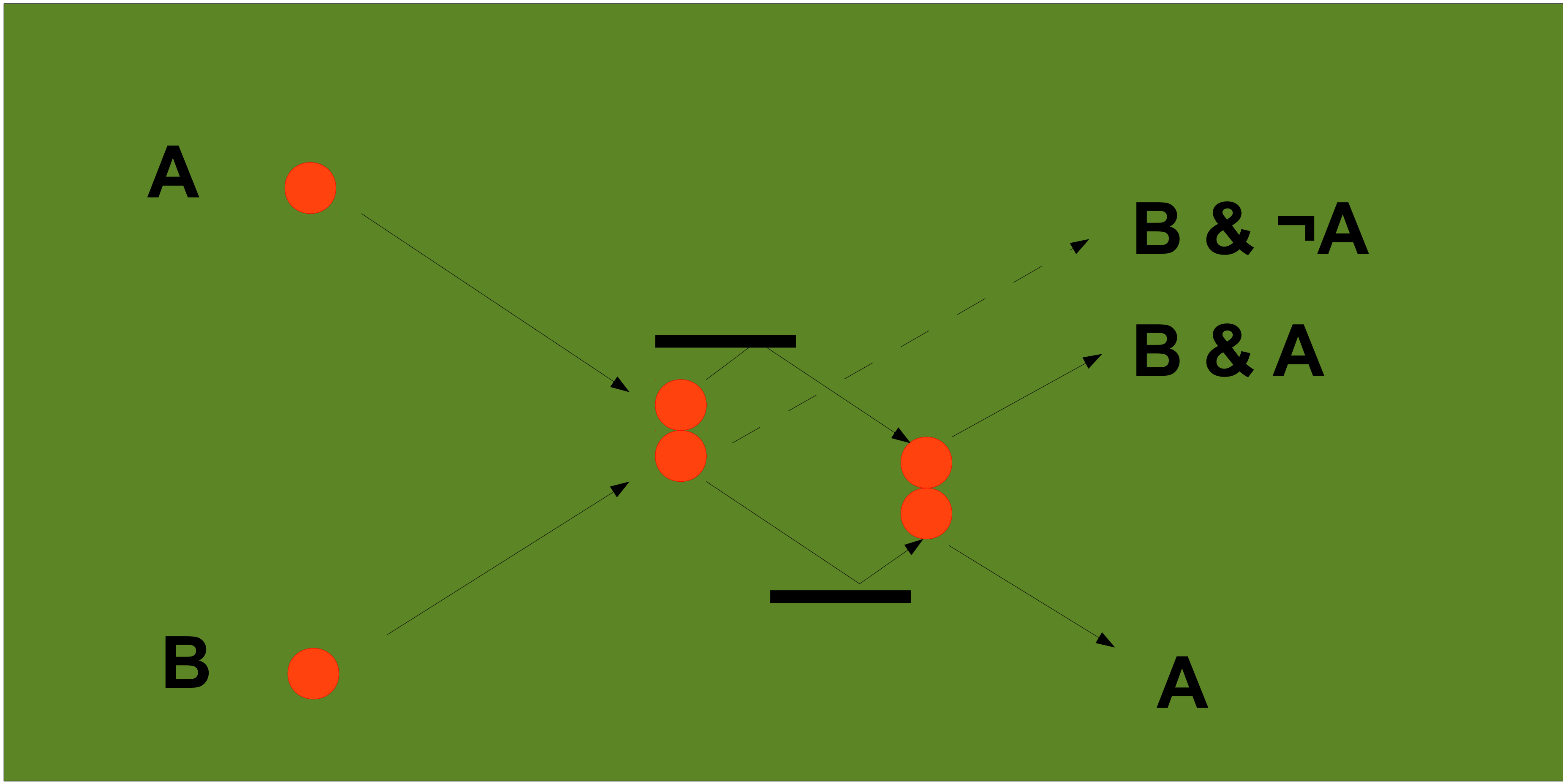
- *Logical Reversibility of Computation* (1973)
- Showed that, in principle, computation is reversible and requires zero energy *if no information is lost*
  - i.e. all state is retained so that we can retrace each step in the computation
- In practice, this means:
  - need a different design for logic gates
  - need to run the computation *very slowly*

# Billiard Ball Computer




Assume no friction, elastic collisions

# Billiard Ball Computer




Use "mirrors" to implement "switching device"
This device is *reversible* because physics is

# Billiard Ball Computer

- Using balls and mirrors, we can implement basic logic gates: AND, OR, NOT

- With a big enough billiard table, we could (in theory) implement a complete computer using a combination of these gates

- BUT …
  - billiard balls don't work in practice
  - normal AND, OR, NOT gates aren't reversible

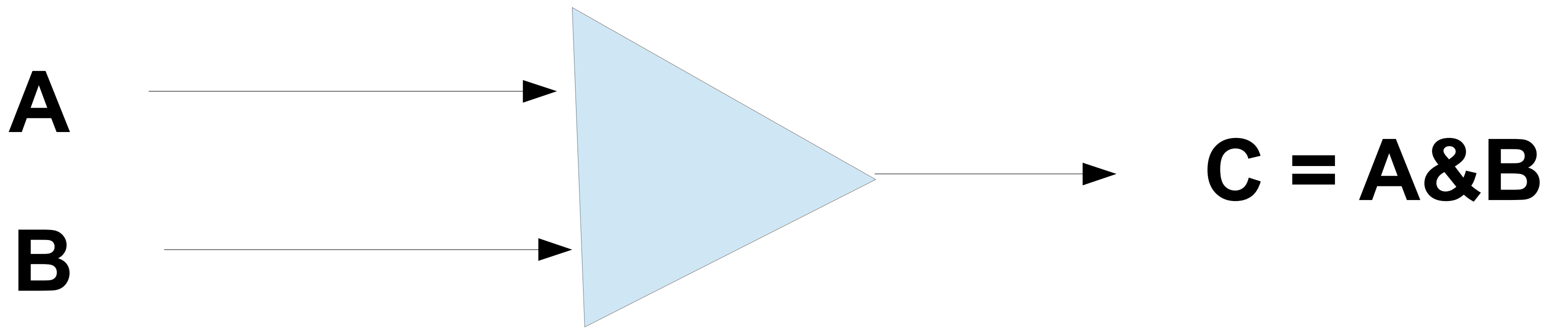
# Why Don't Billiard Balls Work?

- **Thermal losses**
  - friction can't be ignored
  - collisions aren't perfectly elastic

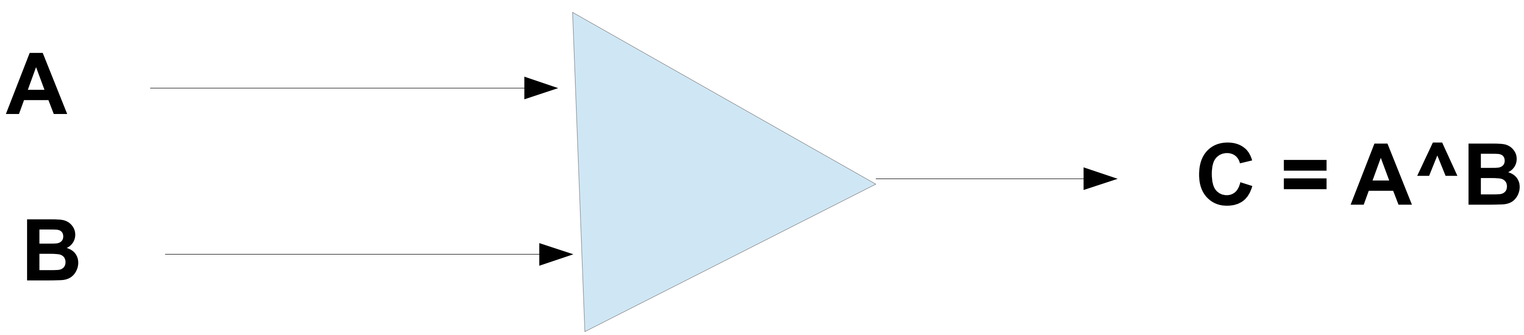
- **Chaotic motion**
  - balls are actually conglomerates of many atoms in various states of vibration
  - can't know their "initial state" perfectly
  - small variations in initial conditional conditions can cause <u>exponentially large</u> differences in final state

# Irreversible Gates

- AND gate



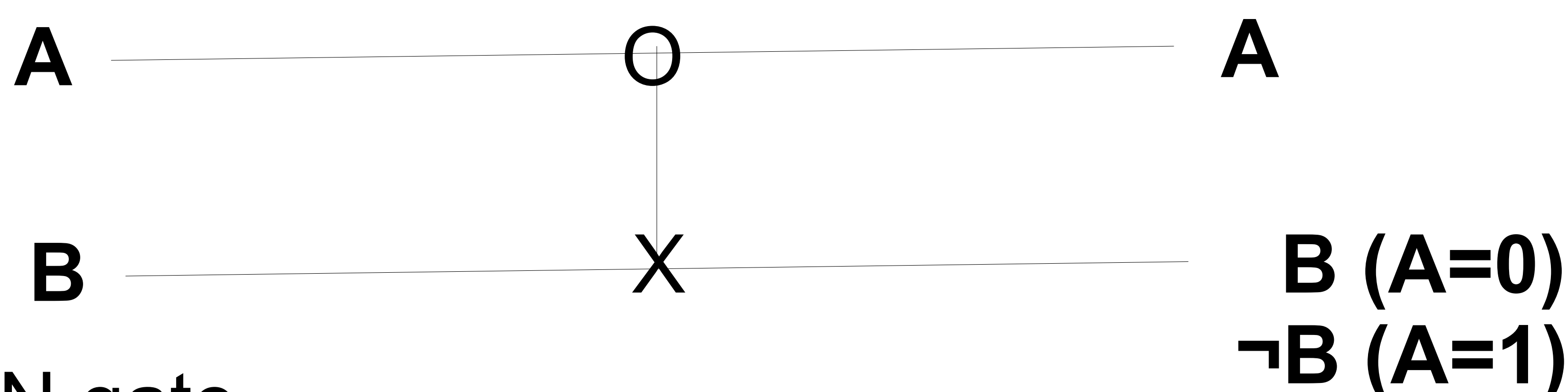

- OR gate



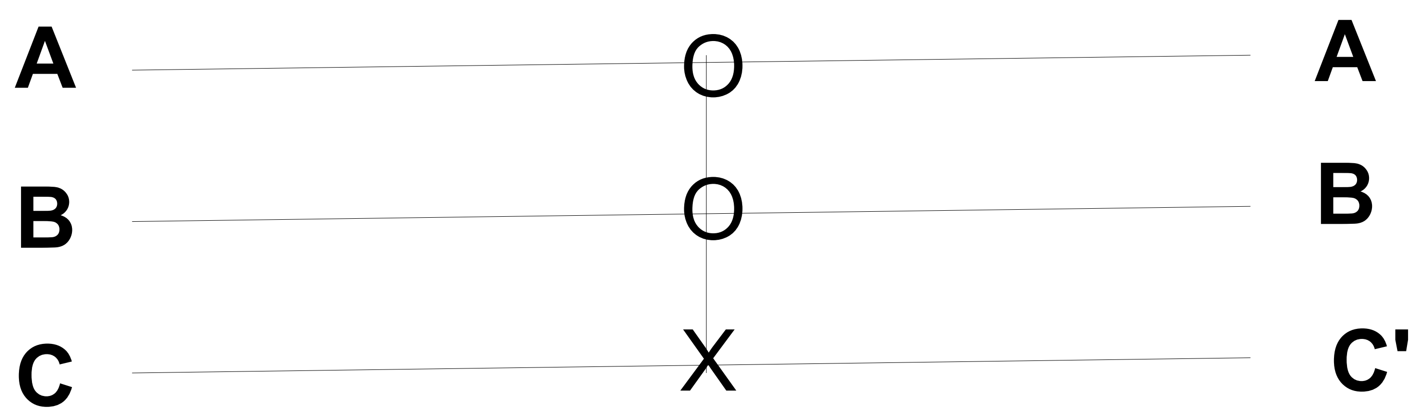

- Can't reconstruct input from output

# Reversible Gates

- Controlled NOT (CN) gate




- CCN gate



# Rules for CN and CCN Gates

- CN is equivalent to XOR (exclusive OR)

- CN followed by CN = no operation,

  i.e. we can reverse the effect of this gate

- All other gates can be built from multiple CCN gates, so that's all we need to build a computer

# Richard Feynman

- *There's Plenty of Room at the Bottom* (1961)

- Introduced the idea of nanotechnology and showed that small devices could be both <u>faster and more reliable</u> than large devices

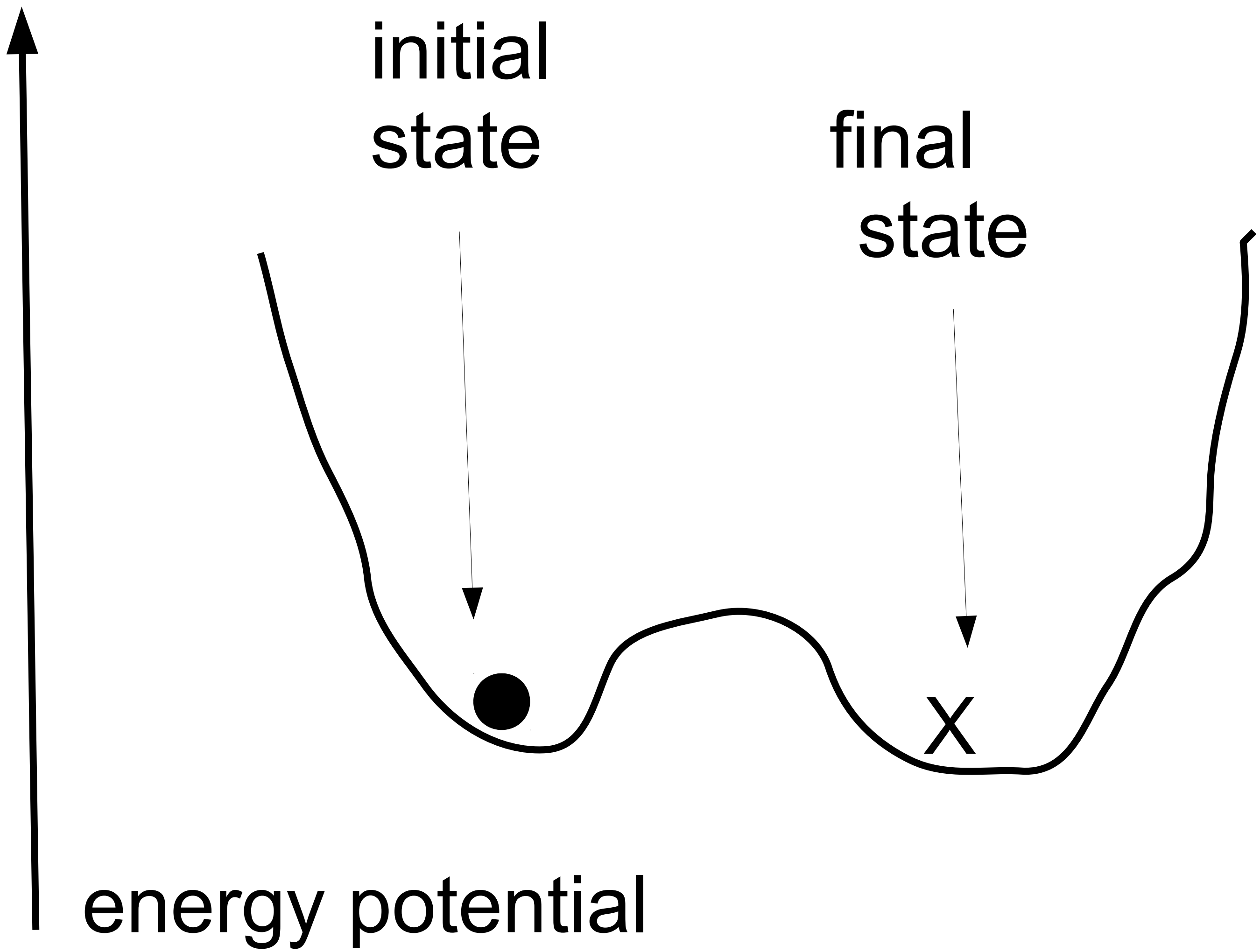
- Led to the “magic of miniaturisation” and Moore's Law

# Two State Devices

- Basic component of a computer can be any two state device, representing one bit, e.g.
  - electromechanical relay
  - thermionic valve
  - discrete transistor
  - embedded transistor in VLSI chip

- Feynman asked “could we use a single atom, a single electron, or something even smaller?”

# Two State Potential Well



# Two State Switching

- To switch from the initial state to the final state, we normally apply energy to enable a “particle” to surmount the potential barrier

- This energy is lost after switching operation, along with the memory of the initial state and an increase in entropy

- Alternatively, lower potential barrier, allow the “particle” to drift across; then raise the barrier

- Can achieve zero energy switch if very slow
  - energy only lost when we reset the device

# Energy Cost vs. Speed

- To drive a computation forward, we have to apply energy:

  energy cost/step = kT log r    (r = rate)

- So we can compute at zero cost, but infinite time, or spend energy to get speed

- Faster computers run hotter!

# Richard Feynman - again

- *Simulating Physics with Computers* (1981)

- Showed that quantum systems cannot be simulated with a classical computer
  - classical computers are deterministic
  - can't generate truly random numbers

- But a quantum computer could be built which would simulate other quantum systems
  - using quantum elements, e.g. electrons, which can exist in a superposition of states

# Two State Device with Superposition

- Electrons (for example) have “spin” and, in a bound system such as an atom, can exist in “spin up” or “spin down” states
  - or use photons polarised “up” and “down”
  - just like a regular two state device

- In the unbound state, they consist of a mixture of up and down states: a **superposition**
  - analogous to harmonics in vibrating strings
  - this is now known as a “**qubit**” (quantum bit)

# David Deutsch

- *Quantum theory, the Church-Turing principle and the universal quantum computer* (1985)
- Showed that quantum computers are universal, i.e. can simulate any possible physical process in a finite number of steps
- A quantum computer could be used to build the ultimate "virtual reality" machine, that could not be distinguished from the real world

# So How Does a QC Work?

- We can build a CN gate from 2 qubits, and more complex circuits using an array of qubits
- The array must be initialised (pgm & data), and then allowed to “evolve” (zero energy computation) according the laws of QM
- There's no way of knowing how long this may last, or whether it will complete, but we can arrange for the QC to tell us via output signal
- We then test whether the result is there

# Quantum "Parallelism"

- During the computation, all states in a superposition evolve independently providing a kind of parallelism
- Certain problems, such as integer factorisation can be sped up exponentially, using Schor's algorithm
- Other "hard" problems can be sped up quadratically
- But only when the machine produces a result; on average, no net performance gain over a number of runs

# The Coherence Problem

- During the computation, all qubits in the array must be maintained in a “coherence”, i.e. in a single <u>entangled quantum state</u>

- But this is notoriously difficult to achieve
  - thermal vibrations can disturb the state
  - measurements will change the state

- Need some kind of “trap” to contain the array of qubits plus cooling equipment to reduce thermal vibration
  - often using lasers for “optical cooling”

# Practical Progress

- 1973  Hans Dehmelt trapped a single electron using an ion trap

- 1995  David Wineland made the first CN gate using trapped ions

- 2005?  Winfried Hensiger created first ion trap on a microchip

- More recent work on error correction techniques

# What's Happening Now?

- Research continues at a number of research centres worldwide
- It's believed that large amounts of money are being spent by national intelligence agencies ...
- … because they want to break classical encryption methods and exploit quantum cryptography … unbreakable transmission of information using quantum entanglement techniques

# Practical Results?

- Factorisation of relatively small numbers using Schor's algorithm has been achieved

- One frequently repeated claim is that Grover's algorithm for searching a list of ***n*** items "will speed up database searching, enabling an item to be found in √***n*** steps" ….???

- Current database search techniques depend on using indexes, enabling an item to be found in ***Log n*** steps, so this seems unconvincing

- **So** ….

# The Moral of this Story is ...

- There has long been a desire to find computing techniques for tackling “NP hard” problems, i.e. faster solutions for algorithms which are currently intractable

- QC is the only known technique which offers a possible solution, but …

- **Don't hold your breath!**

# Sources


- Charles Petzold, The Annotated Turing
- Tony Hey (ed), Feynman Lectures on Computation
- Tony Hey (ed), Feynman and Computation
- John Gribbin, Computing with Quantum Cats from Colossus to Qubits
- David Deutsch, The Fabric of Reality