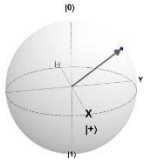
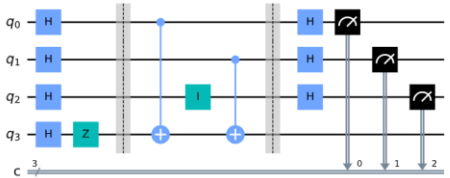
Intro to Quantum Computing
(not covered on exam)

Jon Beaumont



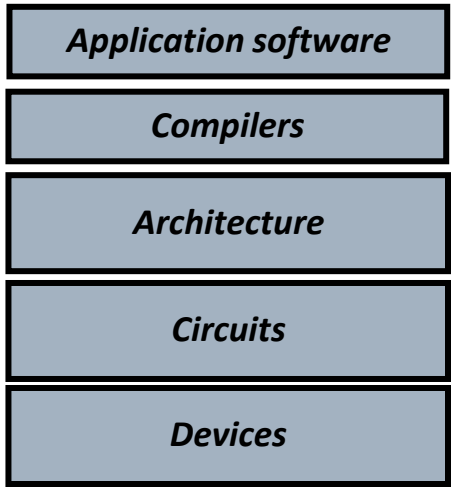


Announcements

- Lab
  - Assignment due Wednesday
  - Canvas quiz by Thursday
  - Meet Fr/M
- Project 3
  - Checkpoint due Thursday – 5%
  - Full project due next Thursday
- HW 3
  - Out later today
  - ~2 weeks to finish

Redesigning the Computing Stack

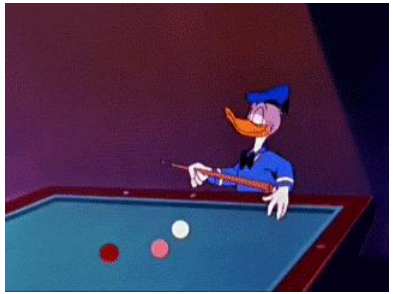
- Many new technologies promise to make bold leaps in computing
  - Neural networks, memristors, photonics, etc.
- Most of these only impact one or two layers of the computing stack
- Quantum computing holds the potential to revolutionize **ALL** layers



Computing Stack

What is Quantum Mechanics?

- Quantum computers use the laws of quantum mechanics as the basis for computation
- Since the 1600s, we've thought the world obeyed the rules laid out by Isaac Newton
  - The world is made of particles
  - Particles have positions and momenta
  - Particles move in straight lines until acted on by a force
  - I.e. the world is an elaborate game of billiards



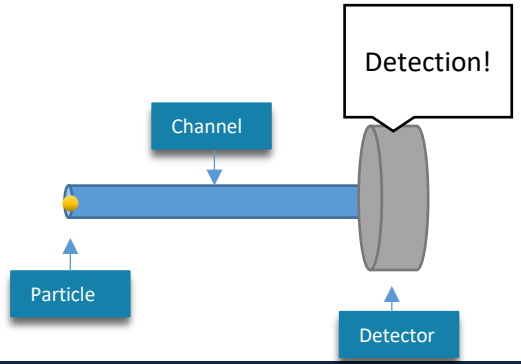
Donald Duck in Mathmagic land

What is Quantum Mechanics?

- 1900s - As scientist probed smaller scales, they discovered that objects act in a very bizarre way when sufficiently isolated from the environment
  - Most evident when dealing with subatomic particles
- Let's devise a simple experiment to illustrate this…

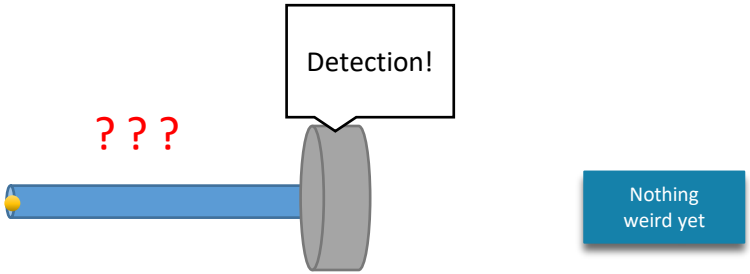
Quantum Mechanics Experiment

- Let's say we have a single particle
  - Anything we can isolate: an electron, nucleus, even bacteria
- Direct this particle down a channel and collide it with a detector
- This is what we observe:



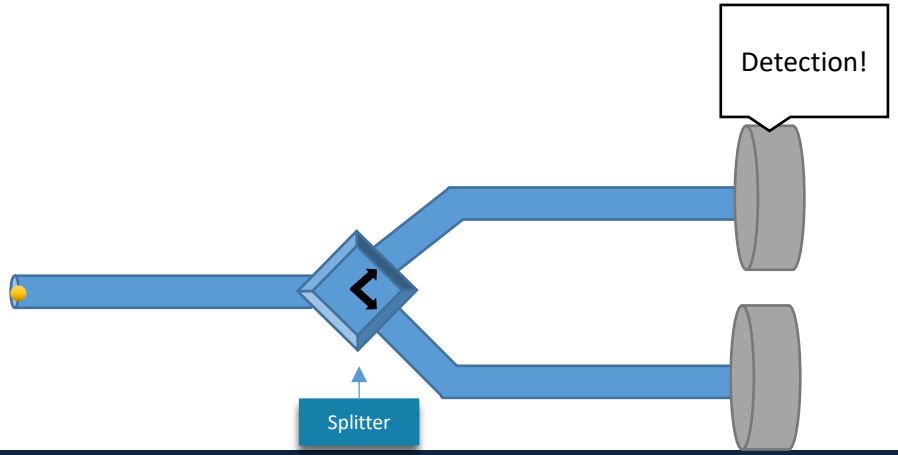
Quantum Mechanics Experiment

- What happens when we completely isolate this particle from the surroundings detection?
  - E.g., no light interacts with the particle
  - We can't actually see the path it takes
  - So from our perspective:



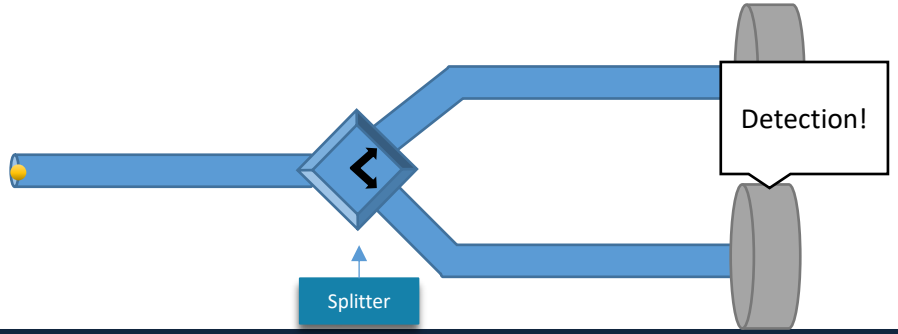
Quantum Mechanics Experiment

- Now let's add a device (splitter) that:
  - 50% of the time directs the particle upwards



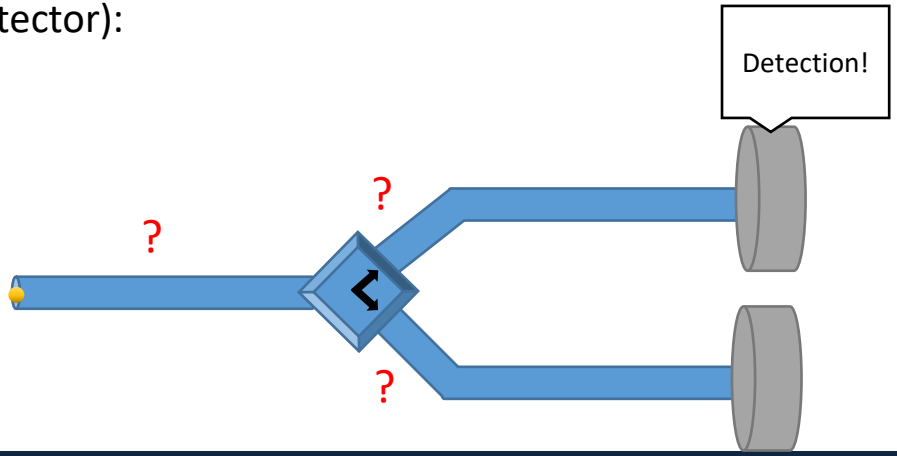
Quantum Mechanics Experiment

- Now let's add a device that:
  - 50% of the time directs the particle upwards
  - 50% of the time directs in downwards



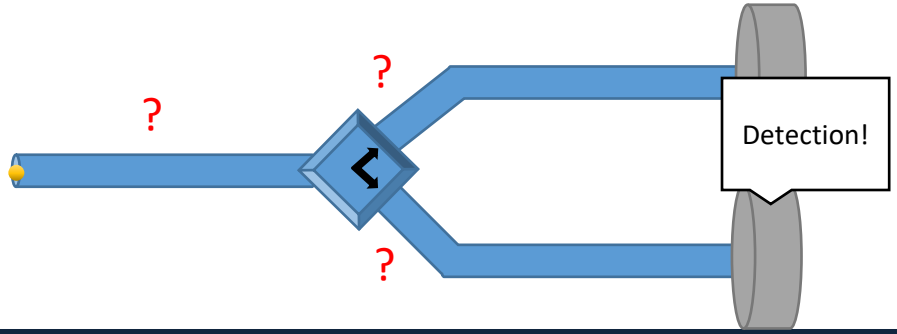
Quantum Mechanics Experiment

- Now let's add a device that:
  - 50% of the time directs the particle upwards
  - 50% of the time directs in downwards
- If we run the experiment many times (without measuring the particle before it hits the detector):



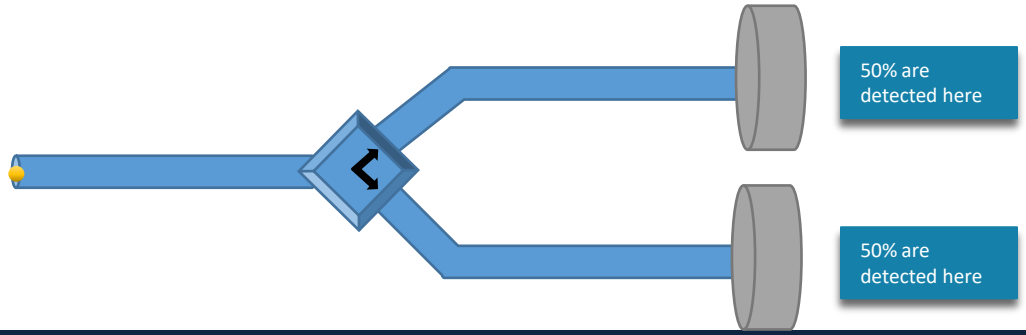
Quantum Mechanics Experiment

- Now let's add a device that:
  - 50% of the time directs the particle upwards
  - 50% of the time directs in downwards
- If we run the experiment many times (without measuring the particle before it hits the detector):



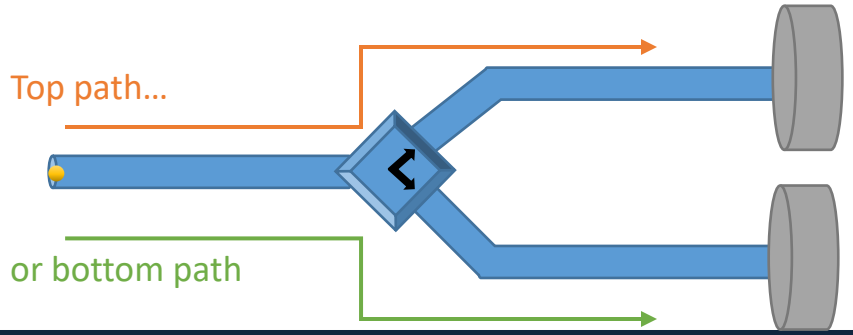
Quantum Mechanics Experiment

- Now let's add a device that:
  - 50% of the time directs the particle upwards
  - 50% of the time directs in downwards
- If we run the experiment many times (without measuring the particle before it hits the detector): we'll see each detector activate half of the time



Quantum Mechanics Experiment

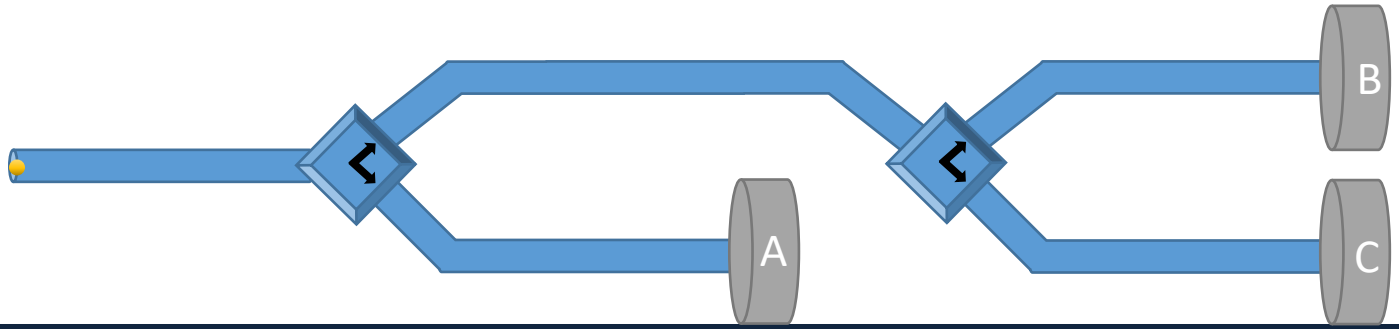
- Scientists in the 1800s wouldn't find this experiment very interesting (yet)
- Even though we don't see the path the particle takes…
  - it most certainly is just taking one of two paths! Right?!
- Let's see if we can find something weird



Quantum Mechanics Experiment

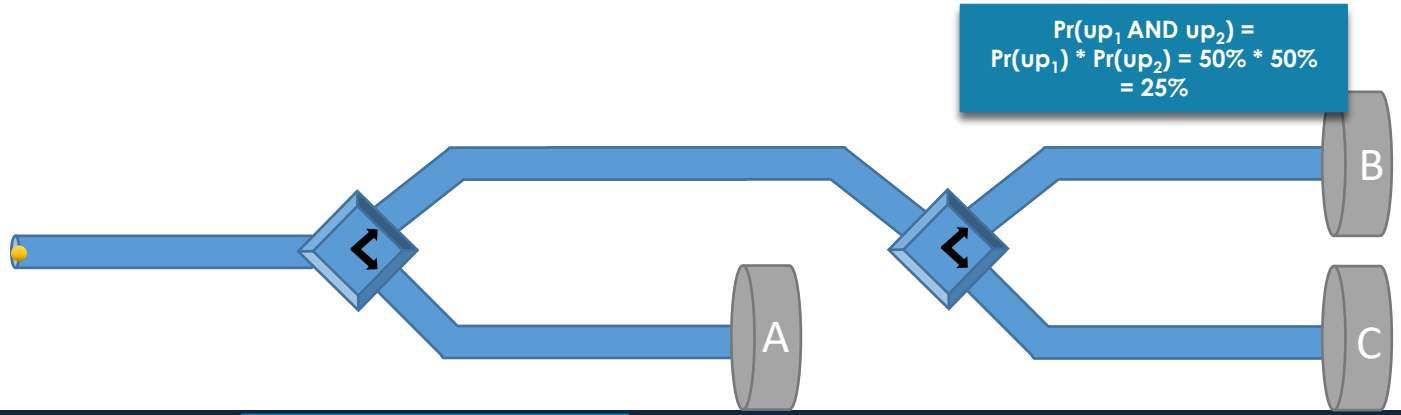
- Now let's modify the top path so that it goes through another 50/50 splitter

Following classical logic & probability, what % do we expect for A,B,C?



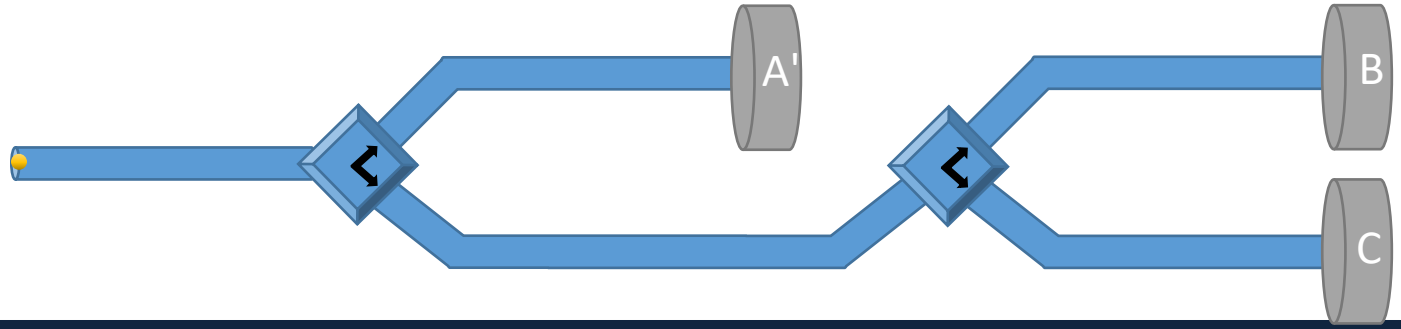
Quantum Mechanics Experiment

- Now let's modify the top path so that it goes through another 50/50 splitter
- Review: probability of 2 independent events happening is equal to product of each probability
- **This is what we observe with experiments, no surprises here**



Quantum Mechanics Experiment

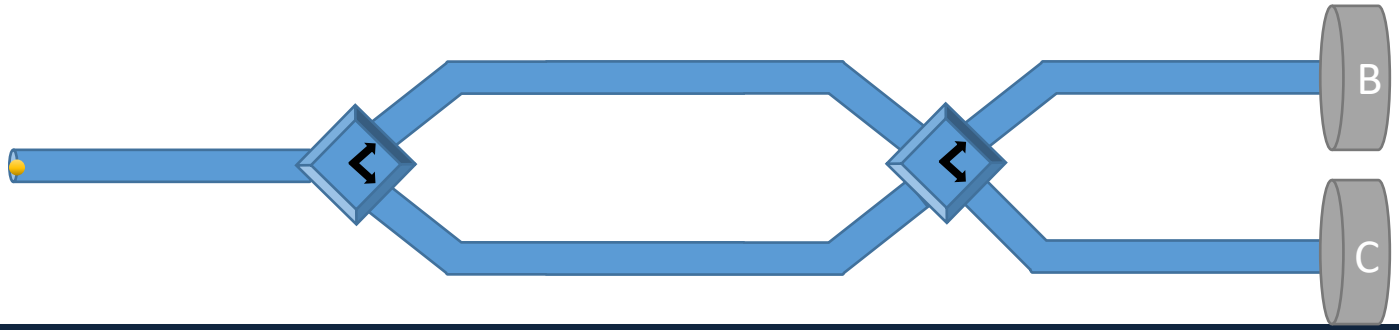
- What if we flip it so the **bottom** path forks again instead of the top?
- Do we expect different results?
- No, we still expect {50%, 25%, 25%}
- **And experiment still agrees with this classical logic**



Quantum Mechanics Experiment

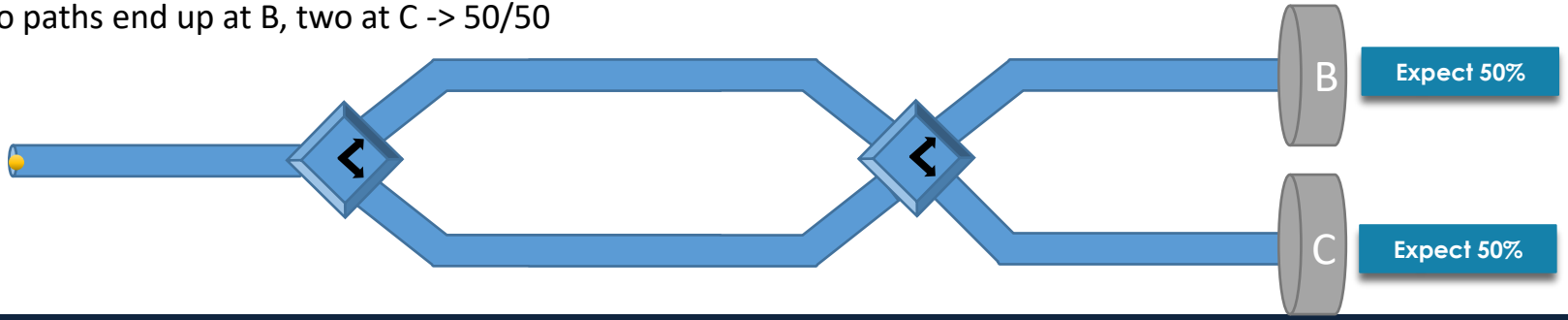
- Now, what if **both paths** merge and go to another splitter?

Following classical logic & probability, what % do we expect for B,C?



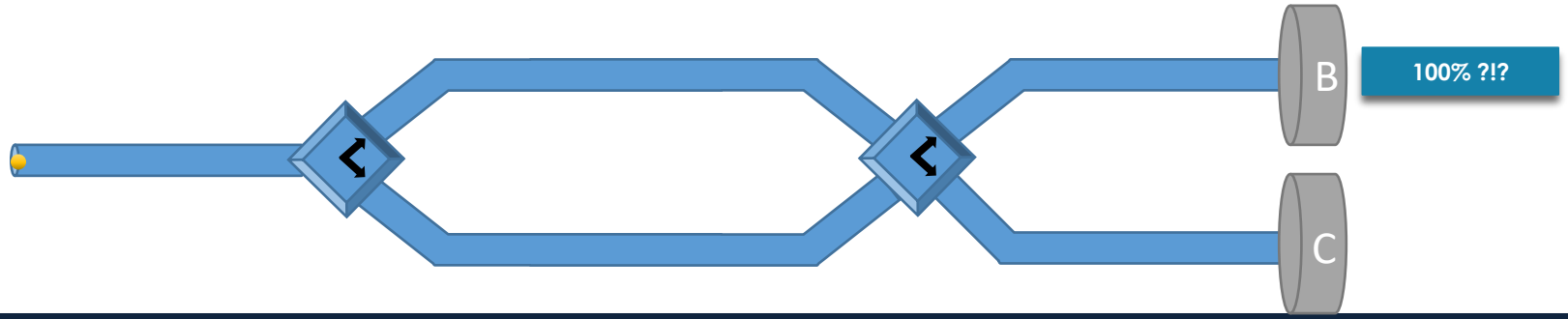
Quantum Mechanics Experiment

- There are 4 possible paths:
  - Up,Up
  - Up,Down
  - Down,Up
  - Down,Down
- All are equally likely, so each one should be 25%
- Two paths end up at B, two at C -> 50/50



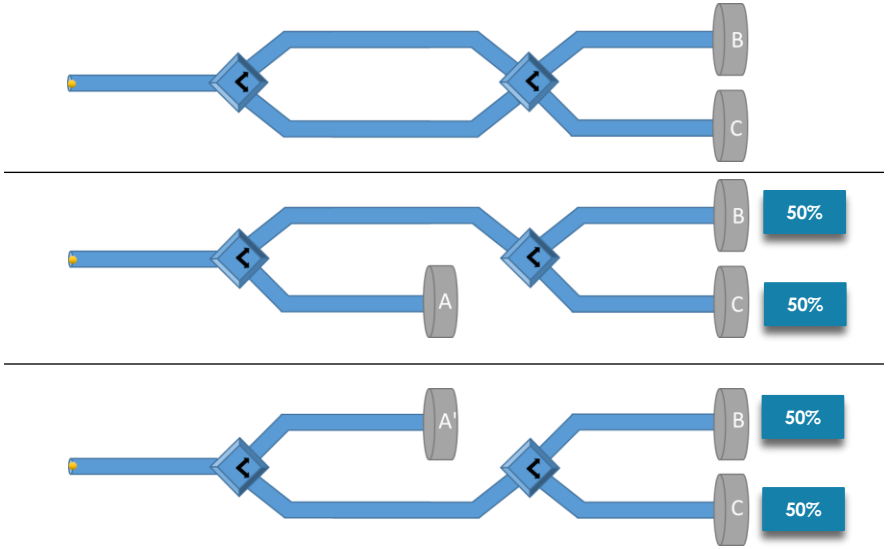
Quantum Mechanics Experiment

- When we run this experiment in real life…
  - This is **NOT** what we observe
- Instead, 100% hit target B



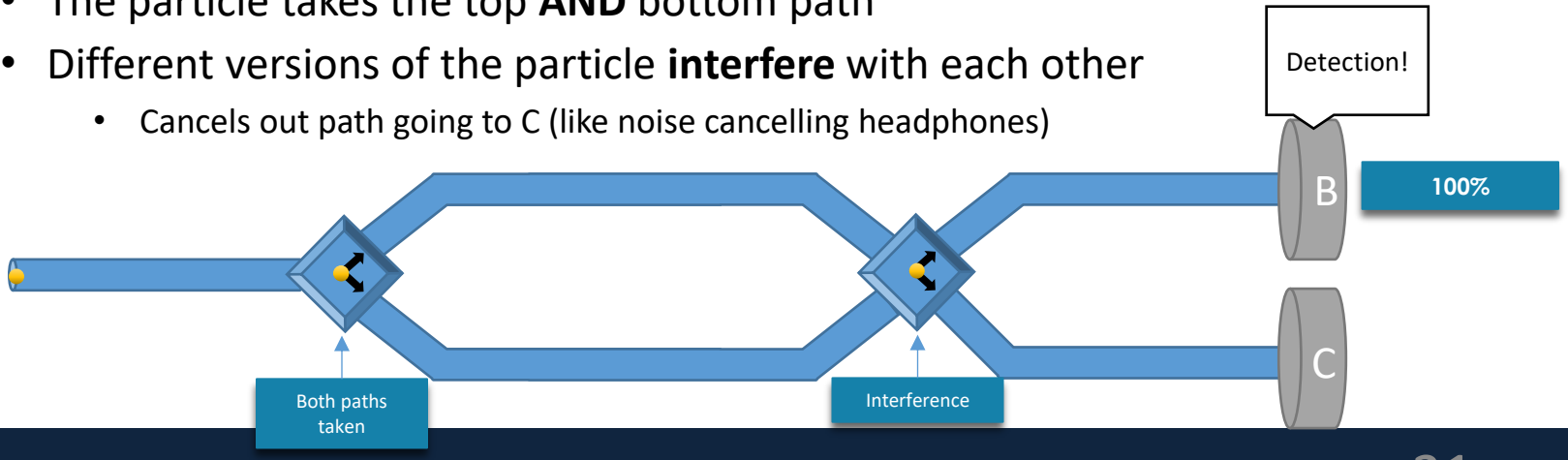
Quantum Mechanics Experiment

- Wait, what?!
- Aren't there only 2 possibilities?
  - Particle goes up at first splitter
  - Particle goes down at first splitter
- Goes **up**? 1st experiment result: splits evenly between B and C
- Goes **down**? 2nd experiment result: splits evenly between B and C
- Either way, B / C should be equal



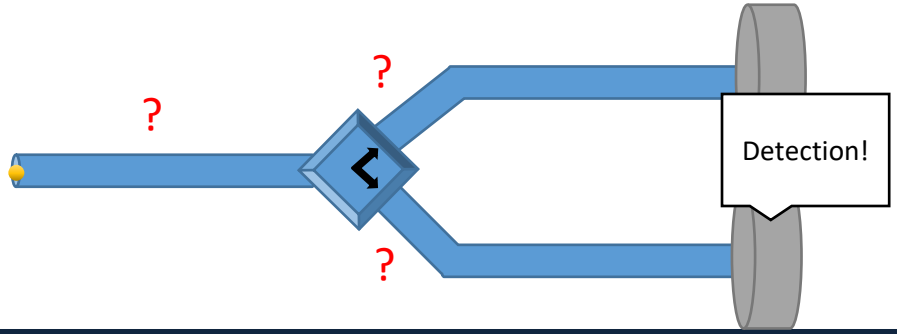
Quantum Mechanics Experiment

- Here's where we went wrong:
  - We assumed particle must take the top **or** bottom path
- Fundamental fact of quantum mechanics:
  - The particle takes the top **AND** bottom path
  - Different versions of the particle **interfere** with each other
    - Cancels out path going to C (like noise cancelling headphones)



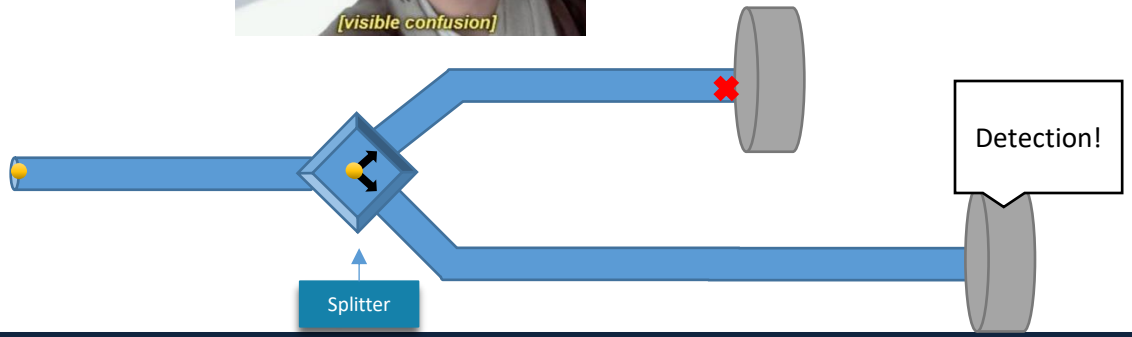
Quantum Mechanics Experiment

- "Yo Teach, this isn't that weird! So the particle is splitting into two pieces, and bumping into each other at the second splitter? Big whoop!"
  - But this can't be all there is to it
- 1st experiment shows particle is only detected on one place
- If the particle split in two smaller pieces, we would expect 2 detections



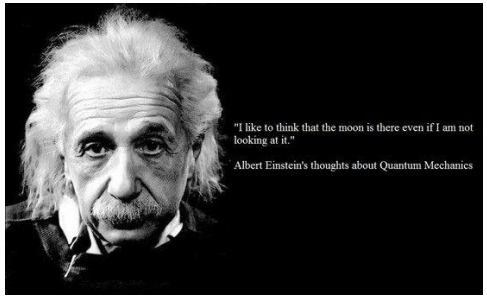
Quantum Mechanics Experiment

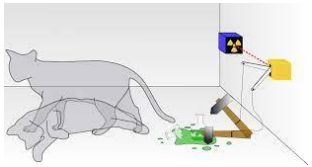
- It's as if the particle takes both paths…
- But as soon as a measurement is made (i.e. as soon as the particle interacts with the environment), one of the versions disappears at random
- … what?!



Quantum Mechanics Experiment

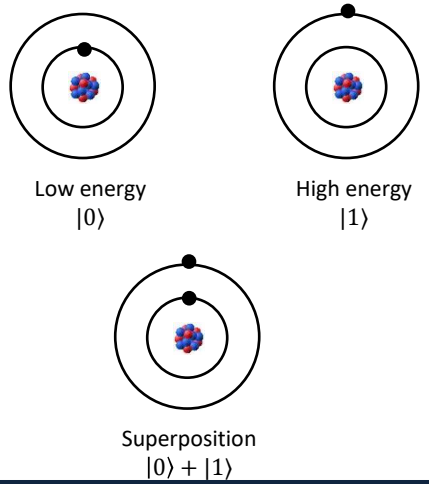
- Can this really be right?
  - Seems absurd: reality works one way until we observe it, and then acts a completely different way?
- Scientists were reluctant to accept it (most famously Einstein), but decades of experiments have verified that this is how the world works
- Principles of quantum mechanics:
  - Properties of systems exist in **superpositions** of states while isolated
    - E.g. a particle exists in two places, an atom has two energy levels
    - A cat is both alive and dead (if we could isolate it entirely… not practical)
  - Systems interacting with their environment **collapses** this states into an observable configuration
  - Which configuration it collapses to is inherently **probabilistic**





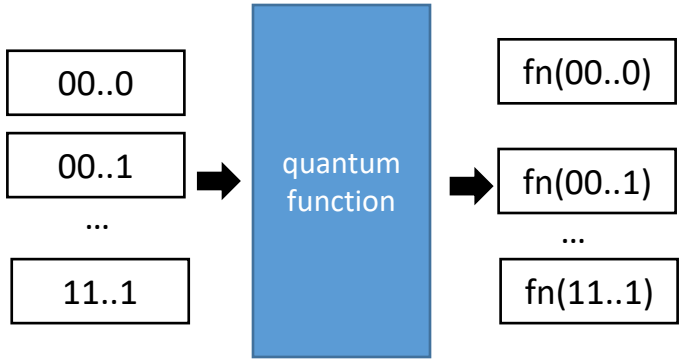
Quantum Computing

- Why are particles relevant to computing?
- 1980s: Richard Feynman and Paul Benioff argued QM could be used for speedups in computation
- Idea:
  - Classical (i.e. non-quantum) computing works on data in the form of bits (0s and 1s)
    - Typically, represented as high or low voltages in a transistor
  - A quantum system could represent one configuration as $|0\rangle$ and another as $|1\rangle$
    - e.g. whether a position is in the top or bottom channel
    - or whether an atom is in a low or high energy state
    - Forms a quantum bit or "qubit"
  - A qubit can be 0, 1 or **both**
- Can we use this to our advantage?



How QC **DOESN'T** Work

- ✓ "If a qubit can be a 0 and a 1, then n qubits could be every value between 00…0 and 11…1 simultaneously!"
- ✓ "Then we could run an algorithm with every possible input simultaneously!"
- ✗ "Then we could learn the output of every possible speedup, giving us exponential speedup on almost every problem!!



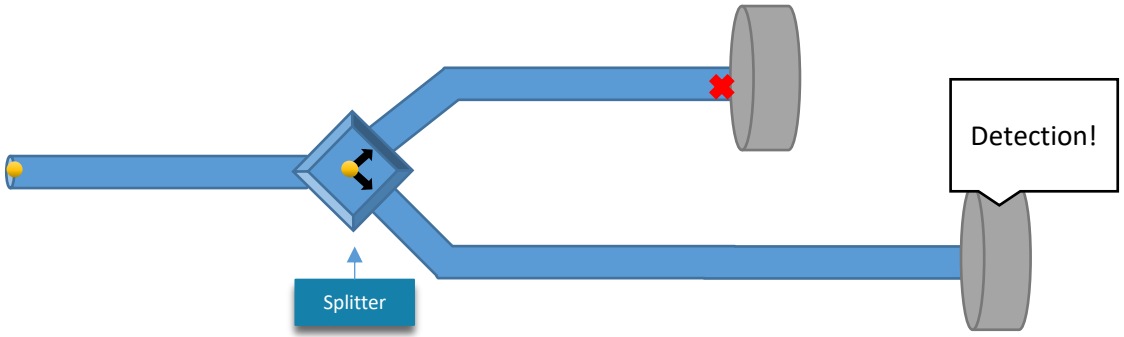
This is how most pop science literature describes quantum computing

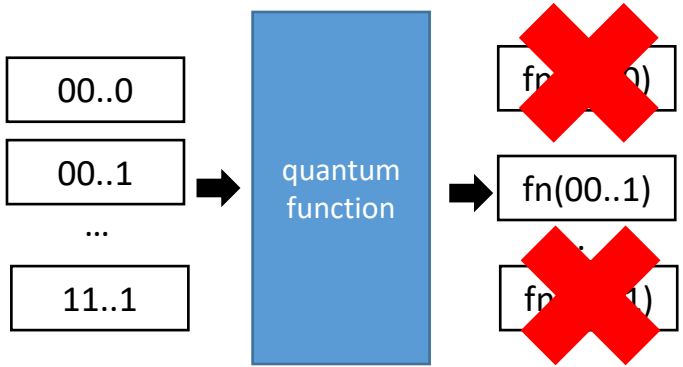
Why is it wrong?

Quantum Computing

- In this experiment, although the particle is in a superposition of top and bottom…
  - the superposition disappears as soon as it's measured
  - We only see it in one location

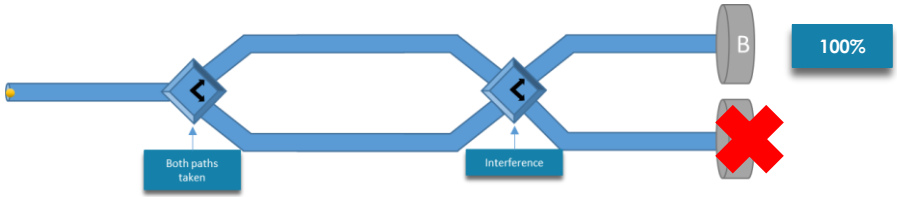
- Similarly here, although we can calculate a function with every possible input
  - As soon as we measure the result, the superposition disappears
  - We only measure one output
  - Why is this better than just randomly choosing input?

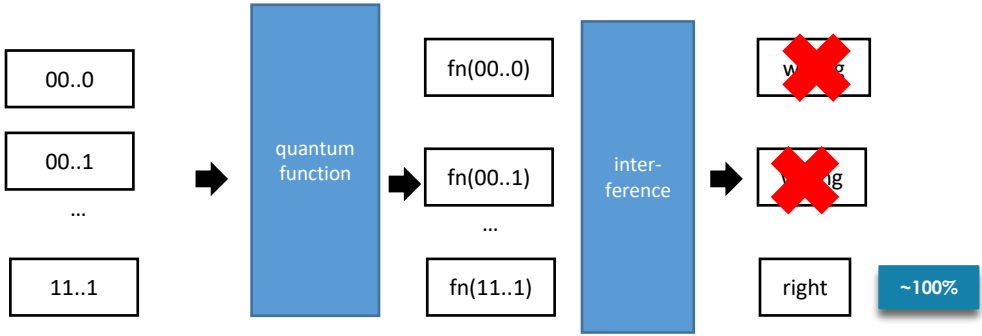




Quantum Computing

- This experiment was only interesting when we had the different superpositions **interfere** with one another
- Similarly, quantum algorithms are only effective if we have some sort of interference step where all possible inputs interfere with on another
- Ideally design such that "good" answers reinforce each other and "bad" answers cancel each other out, making them less likely to measure





Quantum Algorithm Design

- Designing an algorithm to use interference to find something useful is non-trivial
- We've only found a few use cases so far
- But for those cases, quantum computing can be useful in determining some "global" property of a function that would otherwise require several queries from a classical algorithm

Quantum Algorithm Example

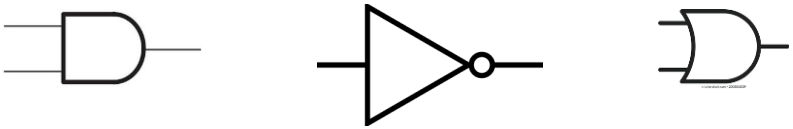
- “Constraint problems” need to find an assignment to a bunch of variables which satisfy several conditions
  - E.g. form study groups so that every student below a B- is matched with at least one student above a B+
- In general, no classical algorithm is significantly better than “guess and check” all possibilities

Quantum Algorithm Example

- "Grover's algorithm" creates a superposition of every possibility
  - Iteratively performs interference until only those that satisfy constraints are likely to be measured
- Instead of checking N possibilities (classical)…
  - Only need to iterate $\sqrt{N}$ times (quantum)

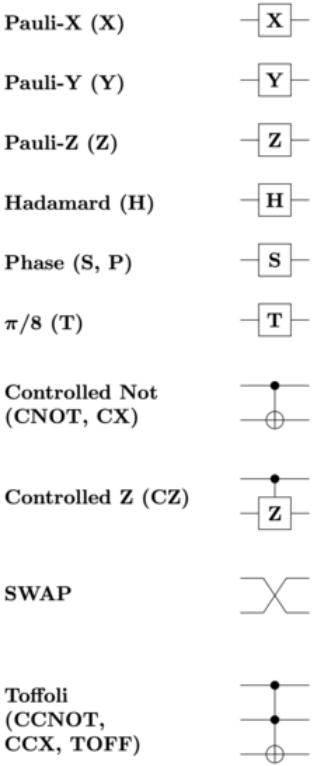
Quantum Logic Design

- Instead of gates that work on 0s and 1s…



- We have gates that work on **superpositions** of 0s and 1s (**MUCH** richer state space, many more gates)

$$|\psi\rangle = \sqrt{.7}|0\rangle - i\sqrt{3}|1\rangle$$



Quantum Computing Timeline

- 1980: Quantum computing proposed as a concept
- 1991: Deutsch-Joza algorithm invented – not practical for anything, but first instance of a quantum algorithm showing exponential speedup over classical
- 1994: Shors algorithm invented, showed large integers could be factored quickly on quantum computer
  - Got everyone very excited in QC… why?
  - This could be used to break cryptosystems (including RSA)
- 1996: Grover's algorithm invented, demonstrates quadratic speedup on huge class of search algorithms
- 2001: Shor's algorithm implemented by IBM to factor 15 using 7 qubits

Quantum Computing Timeline

- Past couple decades: Many companies have invested research in scaling universal quantum computers to larger number of qubits, e.g.
  - Google (53 qubits)
  - IBM (433 qubits – up from 127 last year)
  - Intel (49 qubits)
  - Honeywell (20 qubits – up from 12 last year)

- Estimates are that we'll need 1000s of qubits before anything useful can be done
- All are racing to reach "quantum supremacy"

Quantum Supremacy

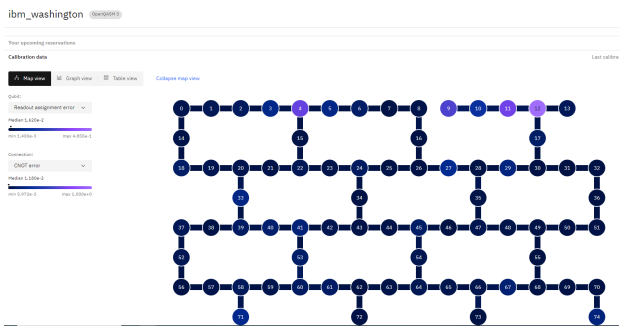
- Quantum supremacy is the point where QCs have been shown to have an advantage over classical computers in some are
  - Doesn't necessarily need to be better at everything
  - Very hyperbolic phrase, "quantum advantage" would be better
- We have not clearly reached it yet
  - Google claimed to do so in 2017, but has been disputed
  - Many other claims are still debated

Is QC going to be "a thing"?

- If you walk around BBB and EECS and ask 10 different professors what they think will happen with QC…
  - you'll get 11 different answers
- Some think the technological barriers are too great, and we'll never see practical QCs
- Some think we'll see a growth like classical computing with vacuum bulbs in the 50s to modern day
  - "It's just a matter of time until we have quantum computers strapped to our wrists"
- A wide spectrum in between
  - Maybe we'll have a few quantum computers on the cloud that a few companies will invest $billions in for very narrow applications, never making a big impact on the average user

What Role do Architects Have?

- For the foreseeable future, qubits will be "noisy"
  - A single photon of light could ruin computation!!
- Systems need to be designed do be "fault tolerant“
- Architects need to understand error rate between different qubits
  - organize them in such a way that “error correction” can be used to undo noise

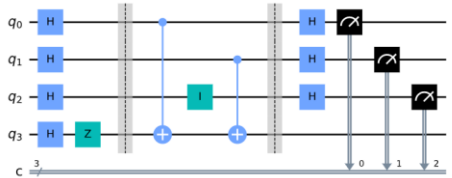


EECS 479 – Introduction to Quantum Computing– W25

- Our class explores different layers of QC with hands-on assignments
- Project 1
  - Program a quantum circuit simulator
- Project 2
  - Design a schedule optimizer using a quantum algorithm faster than any known classical algorithm
- Project 3
  - Design a fault-tolerant computing protocol that allows correct execution even when noise in the environment

Qiskit

- Qiskit is an open-source SDK for designing quantum algorithms
  - Design quantum “circuits” using Python API calls



  - Deploy circuit to simulator or actual quantum hardware on the cloud!

ibm_washington

OpenQASM 3

Your upcoming reservations

Calibration data

Last calibra

Map view | Graph view | Table view

Collapse map view

Qubit:

Readout assignment error

Median 1.620e-2

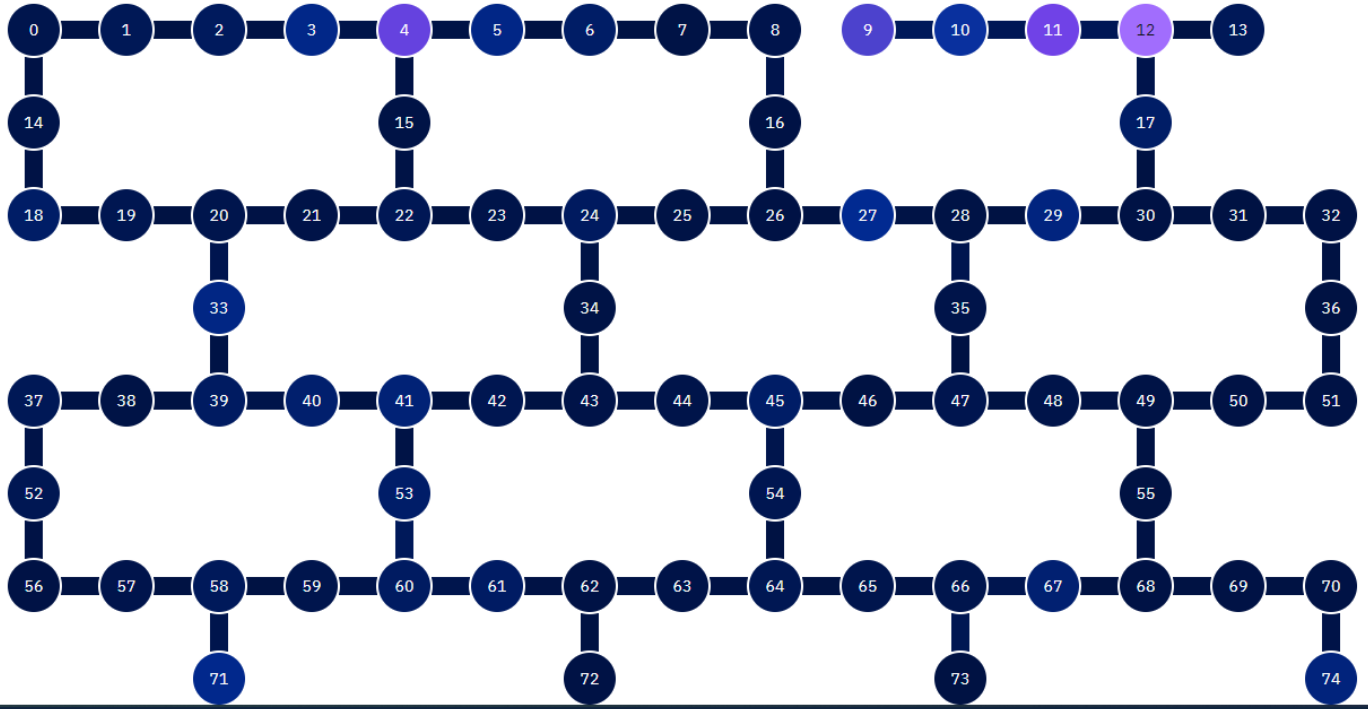
min 1.400e-3 max 4.855e-1

Connection:

CNOT error

Median 1.180e-2

min 5.972e-3 max 1.000e+0



Questions?

Extra Slides

Common QC Myths

- "A quantum computer is like a massively parallel computer, giving us exponential speedup on any parallelizable task"
  - While we can technically compute many solutions simultaneously, we can only measure one result
  - Getting a speedup requires us to be clever about how we setup the algorithm so that one result tells us a lot
  - As best we can tell, we will only get a speedup on a subset of problems, and it won't always be exponential

Common QC Myths

- "Quantum computers can solve previously uncomputable problems"
  - The Church-Turing thesis still holds: anything a quantum computer can do can be simulated by a classical computer
  - But the quantum computer may be much faster!
  - It is true that some intractable (i.e. practically uncomputable at large inputs) can be solved efficiently with QC (e.g. factoring large numbers)
  - Quantum computers can't solve the halting problem

Common QC Myths

- "Quantum computers make encryption impossible"
  - A quantum algorithm ("Shor's algorithm") can break RSA if implemented
  - Other encryption schemes are still secure (although they may need to be scaled up in a post-quantum world)
  - Other post-quantum encryption schemes are being researched

Common QC Myths

- "Quantum computers prove P=NP", or
- "Quantum computers prove P!=NP"
  - We're pretty sure there are intractable problems that won't be improved on quantum computers
  - Doesn't tell us anything about whether problems that can be verified efficiently can necessarily be solved efficiently