CSCI 4140
Quantum Hardware Overview

Mark Green

Faculty of Science

Ontario Tech

Introduction

- There are many things to consider:
  - Qubit representation
  - Computations
  - Coherence
  - Gate Times
  - Error rates
  - Universality
  - Cryogenics
  - Control

Qubit Representation

- Clearly we need to be able to represent qubits
- There are many ways of doing this, the main classification used for quantum hardware
- Need a quantum system that has at least two states, not that hard to find
- Want these states to be well separated, easy to tell them apart
- Don’t want too many states, particularly in the energy band we are using
- Need a mechanism for switching between states

Qubit Mobility

- In some technologies the qubits can move, in other they are in fixed location
- If qubits are in a fixed location they can only interaction with neighbouring qubits
- They can only directly perform gates with these qubits, swap gates are used to move their states to the locations where they are needed
- If qubits are moveable, they can move to where the gates need to be applied
- It is felt that this will give a more scalable architecture

Computation

- In most architectures quantum gates are used for computation, but not all
- Need to be able to change the state of a qubit, this is done by adding energy to the system
- Lasers, microwaves and other forms of electromagnetic radiation has been used for this
- Easy to do for a small number of qubits, can be difficult to scale to large numbers

Coherence

- Quantum states aren't stable, they can change over time
- Coherence is the term used for the useful lifetime of a qubit for computation
- This varies widely with different technologies
  - For some technologies its less than a second
  - For others it can be hours or longer
- This limits the amount of computation that can be performed
- This is one of the major challenges for a useful quantum computer

Gate Times

- The other consideration is how long it takes to perform a single gate
- This varies greatly between technologies, by several orders of magnitude
- None of the current technologies approach the speed of digital gates
- Another consideration is whether several gates can be performed in parallel
- This depends on the amount of cross talk between the qubits and gates

Computations

- The size of the computations that can be performed depends on both the coherence time and gate time
- Basically divide the coherence time by the gate time to determine the size of program that can be run
- Need to add some time to set up the initial data and perform measurements at the end of the computation

Error Rates

- This is the major problem
- Individual gate error rates of 10^{-4} are common, system error rates in the 5% range are common
- Classical computers have error rates in the 10^{-12} or less range, this wasn't always the case
- Two avenues of research:
  - Lowering the gate error rate
  - Error correction – involves many more qubits

Universality

- General purpose computing, may not always be necessary
- If we give up universality we can do much better, more qubits and lower error rates
- Early GPUs were not universal, but we still used them
- If we view quantum computers as a replacement for classical computers universality is important
- If they are just part of a larger computing structure, this may not be necessary

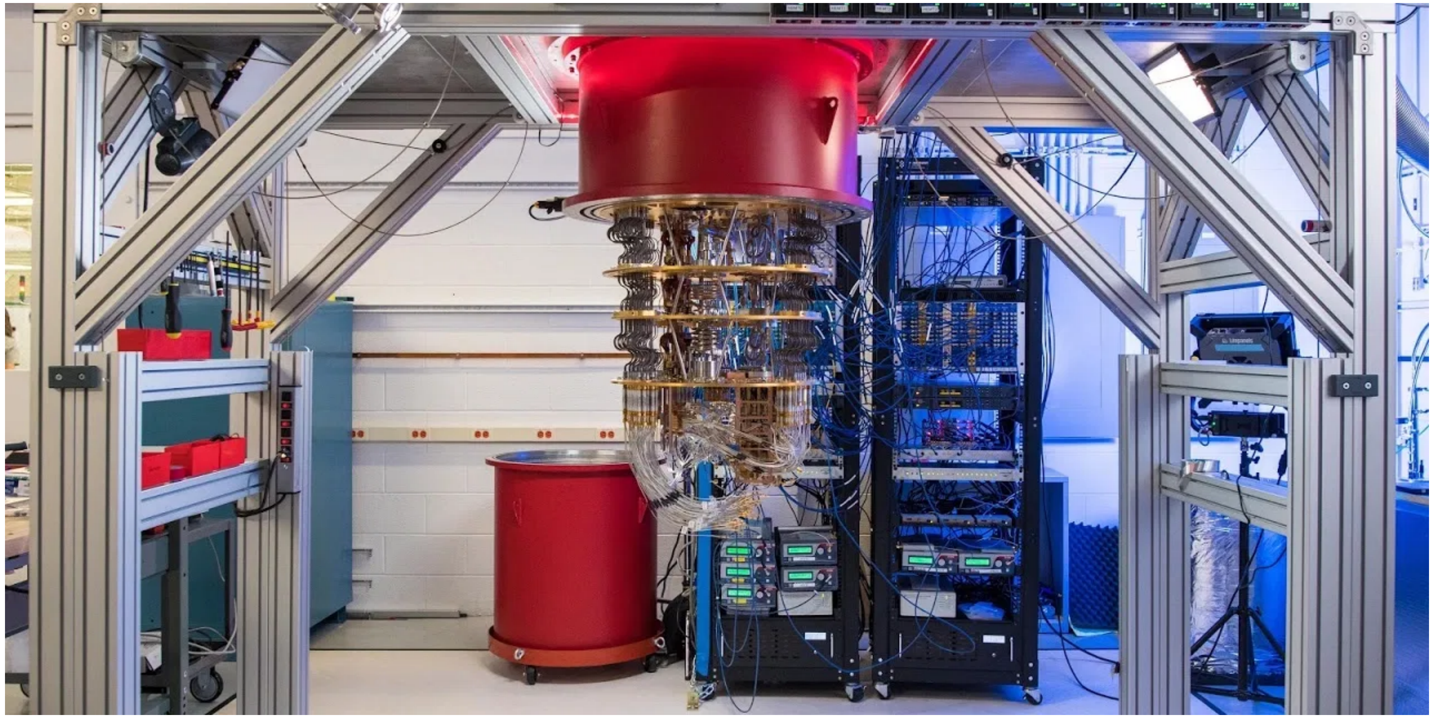
Cryogenics

- Some architectures are cooled to very low temperatures, close to absolute zero, to reduce error rates
- This can be over 95% of the cost of a quantum computer
- Quite large and consumes a lot of power
- This is okay for a mainframe like computer, but wouldn’t work for a laptop
- Also complicates interfacing with the quantum computer
- Technologies that don’t require cryogenics are a big plus

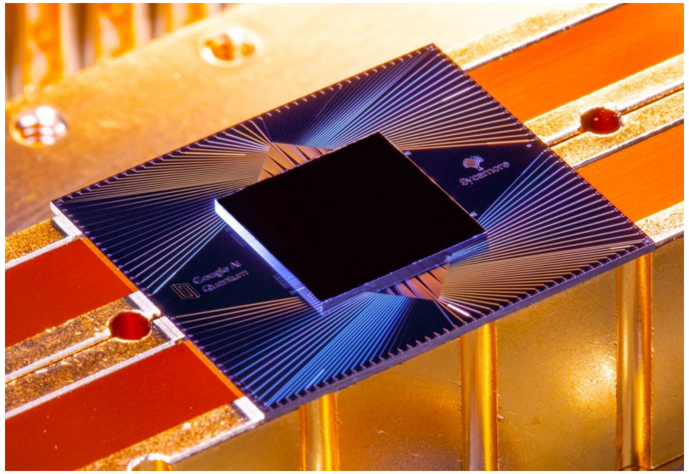
Control

- Getting data into and out of a quantum computer is one of our biggest challenges
- Current systems have classical computers controlling the quantum computer, this works okay now
- As performance increases, classical computers may not be able to keep up, the slow link in the chain
- Need to make quantum computers more independent

Google Quantum Computer - Sycamore



Whole System



Quantum Chip

NISQ

- NISQ – Noisy Intermediate Scale Quantum computers, this is the current hardware generation
- 50 to 100 qubits, no error correction, really not capable of running the classical quantum algorithms
- But, they do seem to be large enough to demonstrate quantum supremacy
- There are applications that can take advantage of these computers, ones that can toleration the errors, some optimization problems

Classical Computers

- Why do classical computers have far fewer errors?
- Use voltage to encode logic levels:
  - 0 – 0 volts
  - 1 – 5 volts
- Note the large difference in voltages, anything under 2.5V is a 0, anything above is a 1
- There is a lot of room or noise, errors are extremely rare
- These are the voltages used off board, used to be the ones used on chips as well

Classical Computers

- The 1 voltage level has been reduced over the years, particularly on chip, two reasons
  - Chip features have gotten a lot smaller, 5V would destroy some of these features
  - Higher voltage generate more heat, require more power, this could cause major damage to chips
- We are pretty much at the point where we can’t drop the voltage lower and stay error free

Classical Computers

- It has taken many years to get to this point, early computers were also very error prone
- Semiconductor technology made current computers possible, reduced the error rate and increased the complexity
- We are now close to the end of this evolution, hard to build faster processors
- Clock speeds have gone down over the past 5 or so years, put more cores on chip and increase chip yield -> reduce cost

Main Quantum Technologies

- The main quantum technologies that people are excited about now are:
  - Trapped ion
  - Superconducting
  - Optical
  - Quantum annealing and special purpose

Quantum Technologies

- Classical computers had the benefit of only one technology: digital logic
- Could concentrate all our efforts on the development of this single technology
- This is not the case with quantum, there are at least four competing technologies
- Effort is divided between these technologies
- Don’t know which one will be dominant, may be several

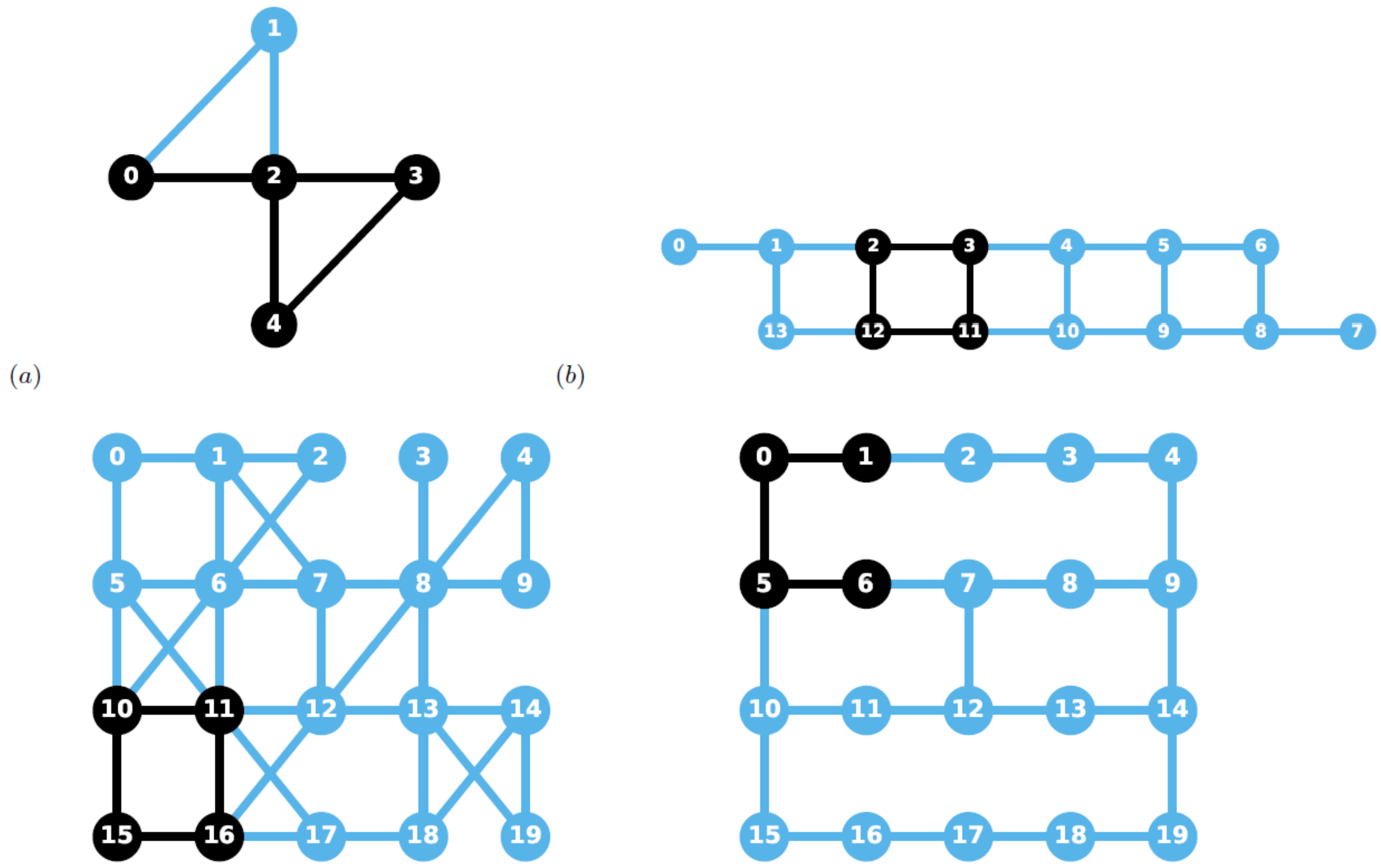
Compilation

- There are a number of topics that impact multiple technologies, we will cover them here instead of multiple times later on
- Each technology implements a limited set of gates, they don't implement all the gates that we've used in our programs
- Theoretically, we only need the Hadamard, S, T and CNOT gate to implement any quantum circuit within a given error limit
- With this set there are circuits that require a large number of gates to implement with low errors, so this isn't necessarily a practical set

Compilation

- The IBM quantum computers implement U1, U2, U3 and CNOT
- Thus any one qubit gate can be implemented exactly with one real gate
- The problem occurs with n qubit gates, we haven’t seen anything above 2 qubits
- Our concern is with implementing two qubit gates, given that the IBM architecture is fixed qubit
- Not every pair of qubits can directly communicate

IBM Quantum Computers



Compilation

- Note how far apart some of the qubits are in these systems, they are not completely connected
- To perform a CNOT between qubits that aren't directly connected, the qubit states must be moved to adjacent qubits using SWAP gates
- Sometimes call this moving the qubit, but it's really moving the state
- Each SWAP gate is implemented as multiple CNOT gates
- Each SWAP gate adds to the gate total, but they must be performed sequentially, which increases the depth of the circuit

Compilation

- One strategy is to allocate the qubits to physical locations in a way that minimizes the number of SWAP gates
- This is a hard graph theory problem, probably need a quantum computer to solve it
- Just starting to develop compilation algorithms, a relatively new research area
- Early results show significant improvements in time and overall error rates

Performance

- With different technologies need some way of measuring performance, some way of comparing different quantum computers
- There are many aspects to performance, not just the number of qubits
- With a short coherence time and high error rate gates can only reliably perform a few gates, regardless of the number of qubits
- A computer with a small number of qubits could easily outperform one with many qubits

Performance

- A computer with high connectivity, or mobile qubits, requires fewer gates to implement an algorithm, faster and more reliable
- How many gates can be applied in parallel?
- Does applying a gate to one qubit effect the state of neighbouring ones?
- Can gates be applied to neighbouring qubits at the same time?
- How good is the compiler?

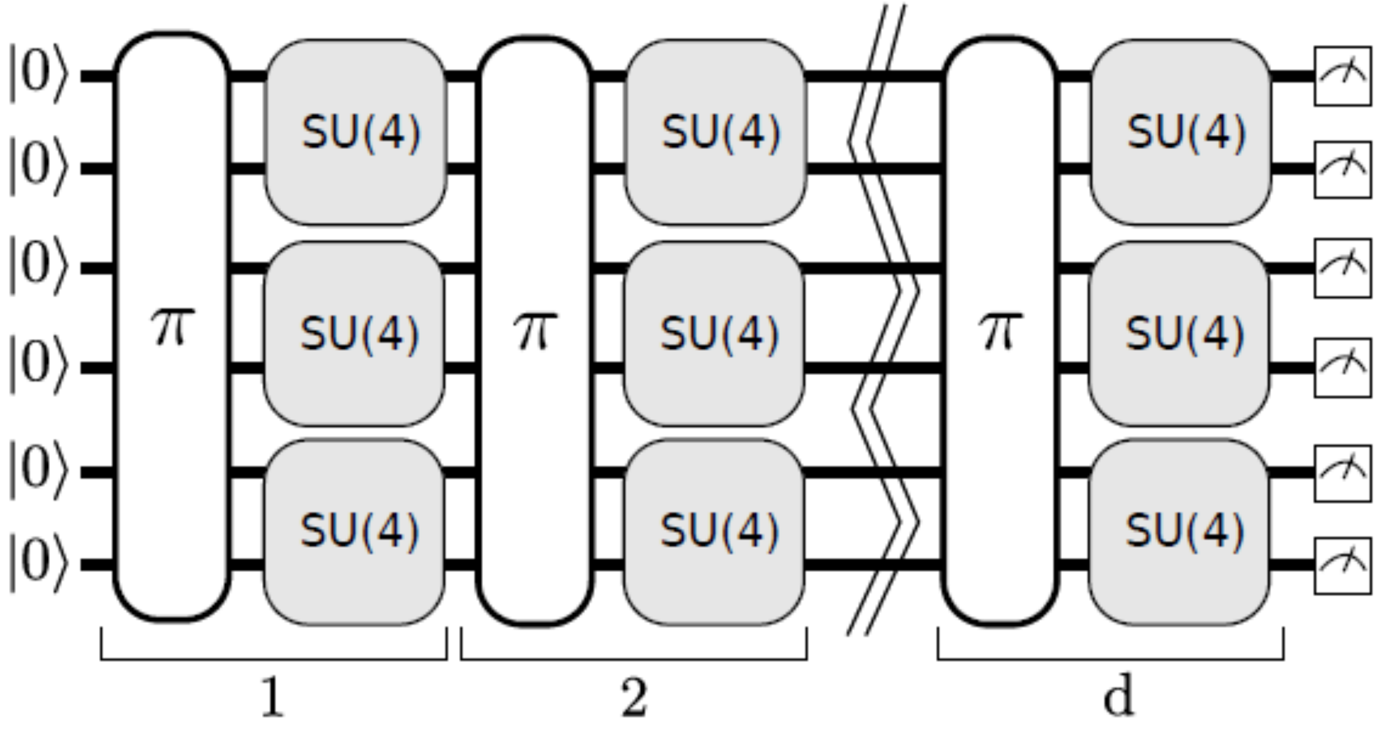
Quantum Volume

- This is the standard metric for measuring the performance of quantum computers
- It provides a single number that includes all aspects of the system including the compiler
- Unfortunately, the description of this metric is quite complicated and the details require a fair bit of mathematics
- Present a high level of view of it, and skip the difficult details

Quantum Volume

- Quantum volume is based on evaluating random circuits of equal breadth and depth
- The breadth, m, is the number of qubits, and the depth, d, is the number of sequential gates
- Random circuits are used since we don’t have a standard set of quantum applications that can be used to measure performance
- The basic idea of the circuit is shown on the next slide
- The set SU(4) includes all of the standard 2 qubit gates

Quantum Volume



Quantum Volume

- Here π is a random permutation of the qubits, this ensures that the gates are performed on random qubits, and no just the ones that are nearby
- In the case where m is odd one of the qubits isn't used
- Each of these quantum circuits can be represented by:

$$U^{(t)} = U^{(t)}_{\pi_t(m'-1),\pi_t(m')} \otimes \cdots \otimes U^{(t)}_{\pi_t(1),\pi_t(2)},$$

- Which is basically the cross product of the layers

Quantum Volume

- We are measuring the outputs of these circuits, but what are we going to do with it
- U is our theoretically exact circuit, not one that has been compiled to run on a quantum computer
- At this point we introduce the heavy output generation (HOG) problem, in high level
- We are measuring the outputs of the circuit, the ideal distribution of these output is, where x is any of the possible bit strings:

$$p_U(x) = |\langle x|U|0\rangle|^2$$

Quantum Volume

- Note that the p_U are probabilities, we compute them for each of the 2^m possible values of x and then order them in the following way:

$$p_0 \leq p_1 \leq \ \dots \ \leq p_{2^m-1}$$

- We then calculate the following:

$$p_{med} = (p_{2^{(m-1)}} + p_{2^{(m-1)}-1})/2$$

- With this the heavy outputs are:

$$H_U = \{x \in \{0,1\}^m \text{ such that } p_U(x) > p_{med}\}.$$

Quantum Volume

- The HOG problem is to generate a sequence of strings where more than 2/3 are heavy
- For an ideal quantum computer the maximum probability of these sequences is 0.85, this is still a statistical problem
- To compute the quantum volume we start with m=d=2, determine if we are > 2/3, if we are we add 1 to m and d and continue
- The last value of m and d where this is valid is used in computing the quantum volume

Quantum Volume

Algorithm 1 Check heavy output generation

function IsHEAVY$(m, d; n_c \geq 100, n_s)$
  $n_h \leftarrow 0$
  for n_c repetitions **do**
    $U \leftarrow$ random model circuit, width m, depth d
    $H_U \leftarrow$ heavy set of U from classical simulation
    $U' \leftarrow$ compiled U for available hardware
    for n_s repetitions **do**
      $x \leftarrow$ outcome of executing U'
      if $x \in H_U$ **then** $n_h \leftarrow n_h + 1$
  return $\frac{n_h - 2\sqrt{n_h(n_s - n_h/n_c)}}{n_c n_s} > \frac{2}{3}$

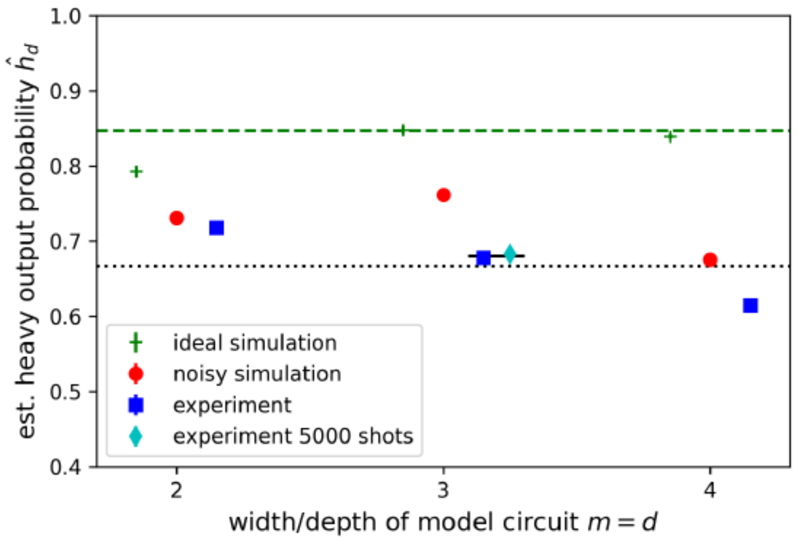
Quantum Volume

- Now we compute the quantum volume in the following way:

$$\log_2 V_Q = \underset{m}{\operatorname{argmax}} \min(m, d(m))$$

- I've skipped a lot of details so you can get the general idea of how the quantum volume is computed
- Computing H_U which we need is exponential on a classical computer, so as quantum computers get larger we will no longer be able to use this metric, or use a quantum computer to compute it

Quantum Volume



Circuit	Tenerife	Melbourne	Tokyo	Johannesburg
$m = d = 2$	0.685 (0.001)*	0.638 (0.006)	0.718 (0.006)	0.711 (0.006)
$m = d = 3$	0.651 (0.006)	0.641 (0.009)	0.682 (0.002)*	0.729 (0.007)
$m = d = 4$	0.516 (0.002)	0.523 (0.002)	0.614 (0.003)	0.664 (0.004)
$m = d = 4\dagger$			0.649 (0.005)	0.699 (0.001)**
$m = d = 5$				0.601 (0.004)

Quantum Volume - Reference

- Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta, *Validating quantum computers using randomized model circuits*, Phys. Rev. A **100**, 032328 (2019). https://arxiv.org/pdf/1811.12926

Laser Cooling

- Okay, this sounds completely bizarre!
- Don't we use lasers to burn and explode things???
- Well, it actually does work
- For several quantum technologies we need to cool the computer to extremely low temperatures, μK, that's micro Kelvin, far less than 1 degree Kelvin
- We can use standard cooling techniques to get down to around 2 or 3 degrees K, but going lower is difficult

Laser Cooling

- Why do we need to do this?
- Quantum states are quite fragile, quite often dealing with individual atoms
- Any amount of heat introduces noise into the qubits and gates
- The more reliable the lower the error rates and the more reliable the computer is
- This is one of the major expenses in constructing a quantum computer

Laser Cooling

- So how does this work?
- Start by working in a vacuum with only one type of atom
- Atoms behave like an ideal gas
- The temperature of this gas is proportional to the velocity of the atoms
- The slower the atoms, the lower the temperature
- The laser produces photons with a precise frequency
- Photons have no mass, but they do have momentum

Laser Cooling

- Atoms respond to particular frequencies, they will absorb photons at these frequencies and ignore photons at other frequencies
- When an atom absorbs a photon, it also absorbs its momentum
- If the photons are travelling in the opposite direction of the atom, this will slow the atom down
- The atom will emit a photon in response to this collision, but these photons will be in random directions
- The momentum of the emitted photons will average to zero, so they won’t effect the atom’s motion

Laser Cooling

- The net effect of this is on each collision the atom will absorb on average half the momentum of the laser photon
- This will cause the atom to slow down, which will lower its temperature
- For a complete system we need 6 lasers acting in the X, Y and Z direction
- The Doppler effect is used to select which laser in a pair will impact the atom

Laser Cooling - Videos

- A real cool introduction to laser cooling (let’s look at it):

  https://www.youtube.com/watch?v=hFkiMWrA2Bc

- A much more detailed description:

  https://www.youtube.com/watch?v=rrNTGJ_-J4I&feature=youtu.be

- You can look at this one yourself later

Summary

- Examined some of the high level issues in quantum computer architecture, the things that must be considered
- Briefly introduced quantum compilation
- Examined the problem of measuring the performance of quantum computers, Quantum Volume
- Examined laser cooling
- Next examine the main architectures in detail