Seminar 2

Created	@September 27, 2022 1:09 PM
Class	Quantum Computing Fundamentals
Type	Seminar
Reviewed	☐
Date	@September 25, 2022
Materials	

Content Covered:

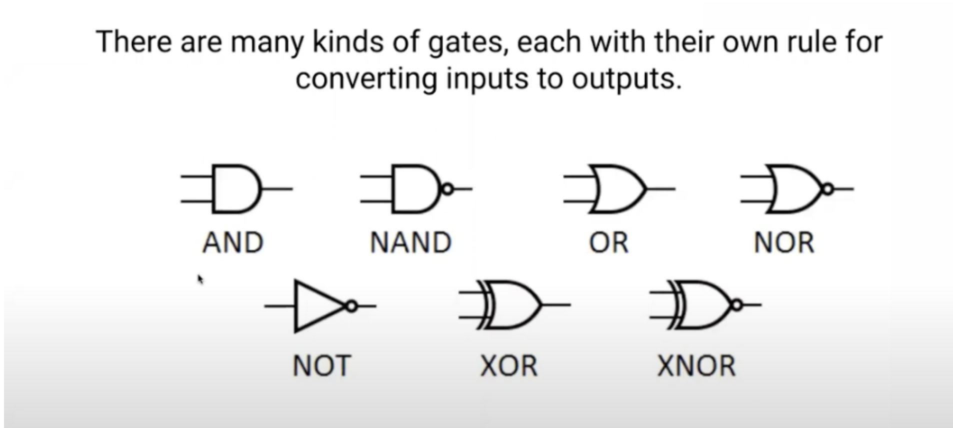
- Classical Stack
- Quantum Stack
- State of the Quantum Stack

Review:

- Quantum computers ≠ Classical computers
    - Q-computers are built different and can help with searching algorithms
    - Q-computers have a wide variety of applications that enhance the current proccesses
- Q-computing is part of Q-sensing and Q-networking (A whole community of applying principles of quantum mechanics)
- How do we build and use a quantum computer?
- Quantum Stack:
    - framework relating all the different concept in this course

    - Currently being developed
    - Compared to like looking at pieces of an entire puzzle
- Classical Stack:
    - Gives us a framework to understand the q-stack
    - Categorizes all layers of a computer’s operation
    - While using our computers, you only see the highest level of the stack
    - Exploring the layers of the classical stack:
        - Applications
        - Classical Algorithms and Protocols
        - Classical Circuits
        - Classical Gates
        - Classical Bits
    - Classical Bits (Most Fundamental):
        - Bits = language of classical computers
        - bits can be 1 or 0
        - everything stored, displayed or computed in computers is stored in 0s or 1s
        - This specific language is called “binary” (only two options)
        - Bits can be compared to lightbulb switches
            - Nothing in between in classical bits (either on or off)
        - How do bits represent complex information?:
            - brightness of each pixel in an image is represented in teh computer by 8 bits
            - 00000000 - completely dark (black pixel)
            - 11111111 - completely bright (white pixel)
            - Values in between - gray pixel
        - What are bits physically?:

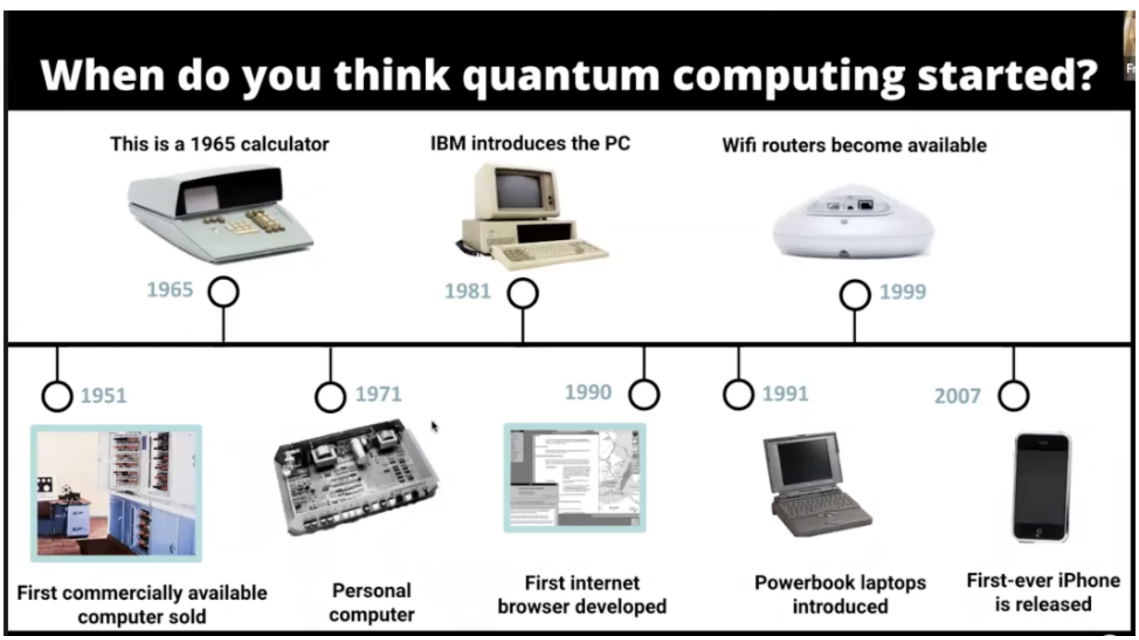
        - Classical bits are pulses of electricity
        - When electricity is flowing - 1
        - When electricity is absent - 0
        - There is no in-between state in classical states
    - Why do our laptop batteries eventually die if they aren't connected to power?
        - If you can't recharge electricity in battery (which dead), the bits that are "1" cannot be powered and thus the bits will all be "0s" and thus no complex computations and applications can be run
- Classical gates (Next level):
    - Without gates, we would only have bits
        - This would mean it would be static
    - In order to make computers dynamic
        - Gates convert "bit" inputs into outputs (also as bits) which are modified
    - Gates change or manipulate "bits" to enable computation
    - Types of logic gates used in classical computing:



    - Every single on of these gates takes an input and produces an output each performing a specific operation on the input
    - The NOT gate for example outputs the opposite value of the input

            - Ex: 0 bit is converted to a 1 bit
        - Example: Inverting an Image:
            - Images are fundamentally represented by a large number of pixels:
                - To invert this, all the white parts turn black and vice versa
                - Using the NOT gate, we can apply this to all the pixel bit string encodings to create an inverted colored version of our image of interest
        - How many gates do you think are in the computer you are using right now?
            - ~ 450,000,000 gates (all of these contained within our portable laptops)
            - Physically gates are transistors. They covert inputs into outputs in the form of electrical pulses (voltages)
            - Gates are made up of 1 or more transistors working together
            - Voltage can be high (1) or low(0)
            - Transistors these days are approaching the quantum level since they are getting smaller and smaller
    - Classical Circuits:
        - Circuits = many gates
        - Gates operate on bits, while circuits operate on gates (which operate on bits)
        - Circuits in the real world:
            - They are all centered around combining gates in clever ways to do intricate computations
            - Computer chips = complicated circuits
                - Crucial for all classical computers
    - Classical Algorithms and Protocols:
        - Algorithms are step by step instructions computers use to perform a complex task
        - Group of interacting circuits = algorithms

        - Similar to an instruction manual that are broken down in a step by step process
        - Start with basics and build into a complex item when algorithm is followed **correctly and in the right order**
    - Applications:
        - **High level that users of computers interact with on a day to day basis**
        - Examples:
            - TikTok, Netflix, Youtube, Discord, Zoom, Slack, etc…
            - These applications that the users interact with are comprised of the fundamentals of the stack that are below it
        - Bits - a way that computers record information through through presence or absence of electrical pulse
- Quantum Stack:
    - **History of Classical Computers:**

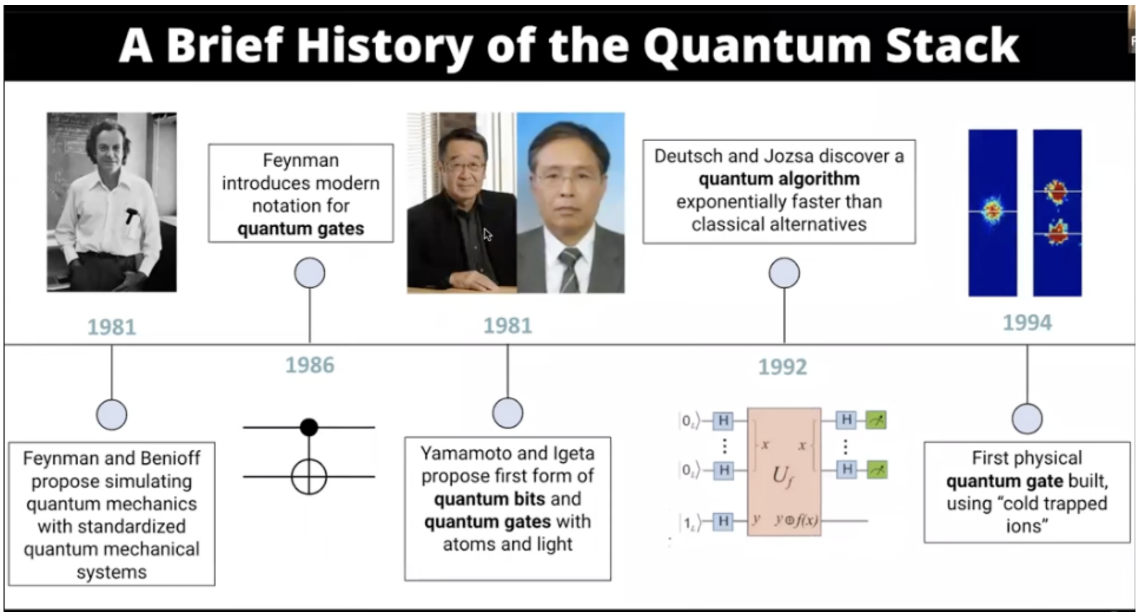


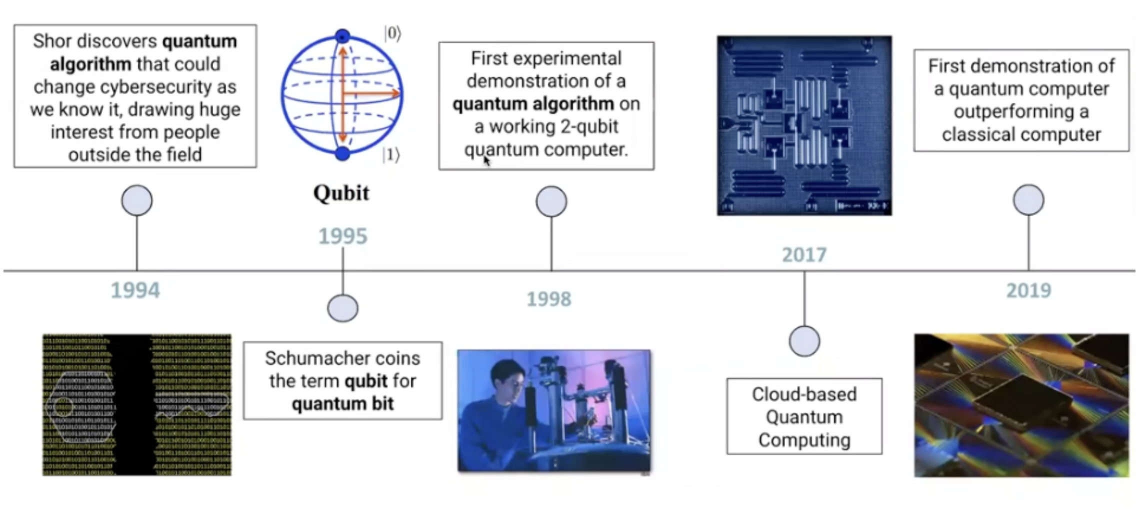
    - **History of Quantum Computers:**
        - In 1981, the idea of Q-computing is introduced

- In 1994, it was proven that q-computers could solve practical problems faster than classical computers
- In 1988, the first q-computer was built

◦ **A Brief History of Q-Stack:**

- Remember: Q-computing is an application of quantum mechanics



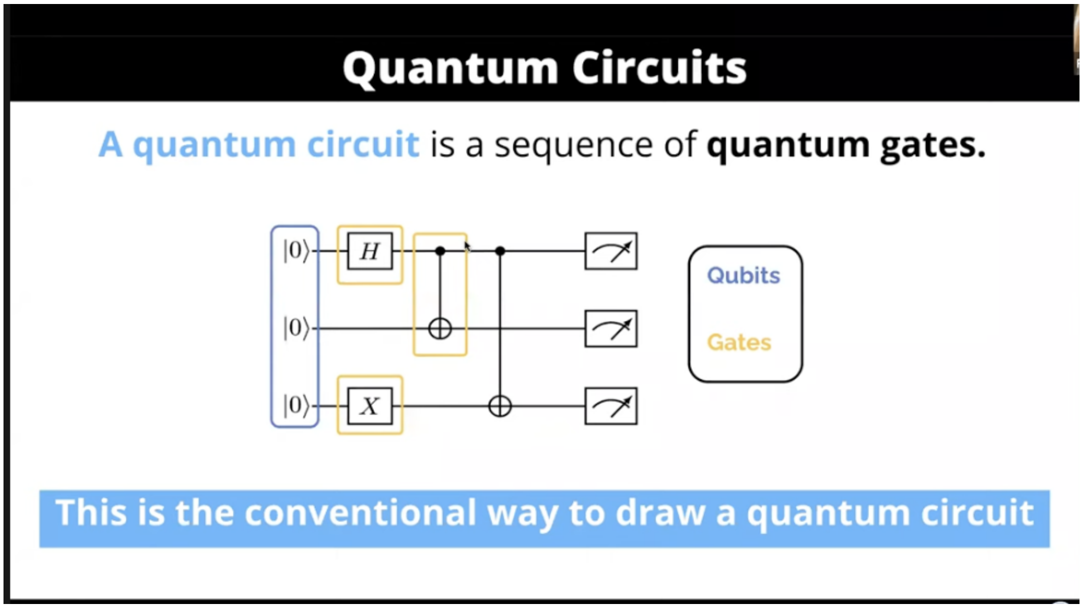


- A big Breakthrough:
    - The discovery of "shor" algorithm by Peter Shor

        - Shor algorithm was important since it demonstrated in the quantum case that we would be able to do significantly better than we can do in the classical case
            - implications in RSA cryptography
            - A quantum computer can do integer factorization infinitely faster than with a classical computer
            - RSA cryptography is used currently to keep all our data secure today
            - This led to military and government interest in q-computers
- Quantum inspired classical algorithms
    - People take ideas form quantum algorithms to improve classical algorithms in systems
- **THE QUANTUM STACK:**
    - Applications
    - Q-Algorithms and Protocols
    - Q-Circuits
    - Q-Gates
    - Qubits
- Similar high level structure between Q-stack and classical stack, yet key differences do exist
- How Would You Use a Q-Computer?
    - Use an application without worrying about the underlying details
    - Scientist wants to simulate a molecule - provide input to a q-computer - q_computer shows you the output
    - YET, that high level of abstractions **does not exist yet**
    - Q-stack has been worked on for around 20 years so far (q-computing is much more recent)

        - As a result, we will study the underlying levels of q-computers since we don't have the hardware to support the high level technology

- Quantum Bits - "Qubits"
    - A qubit is a fundamental unit of information on a q-computer
    - Qubits can be in a combination of these two states (0 and 1 at the same time)
        - Relates to the idea if superposition in q-mechanics
    - Qubits physically can be many different things
        - super-conducting technology
        - trapped ions
        - Photonics
        - NV Centers
    - One physical implementation hasn't been discovered as the superior option yet
    - Comes under the idea of q-hardware
    - Not all q-computers are the same (built in different ways)
- Quantum Gates:
    - Perform manipulation on Qubits
    - Many different types of Q-gates
- Quantum Circuits:
    - A sequence of q-gates forms a q-circuit
    - enables more complex computation
    - Image that is usually drawn:



- Wires represent the evolution of the system over time
- Quantum Chip - complicated Q-circuits
- **In this class we will learn how to interpret and code quantum circuits using Qisikit**
    - Qiskit is a a programming language for q-computers designed by IBM
    - You will get the chance to design q-circuits and q-algorithms and program/implement them in q-computing facilities

▪ Quantum Algorithms and Protocols

- Use q-circuits for each step in a series of steps with the goal of performing a computation
- Example: Grover's Algorithm
    - Gives you a speed up for an unstructured search
    - A normal computer would require O(N) steps, while grover's algorithm on a q-computer will allow a quadratically faster speed O(sqr_root(N))

▪ Applications:

- We are just beginning to explore this field since it we are relatively new to the field
- Quantum Machine Learning is an area in which the instructor is working on right now hoping that in the future q-computers will be more complex to allow for such applications

The Quantum Computing Landscape:

- IBM Quantum, ZAPATA, Microsoft, Google, Rigetti, IonQ, XANADU, PsiQuantum, PENNYLANE
- Quantum computing startups is a growing area
- IBM is building the super conducting q-chips and systems for the lowest levels and have Qiskit programming language at the top
- Google similarly has superconducting qubit technology and Cirq programming language
- Q# is the programming language for the high level
- Q-stack is growing exponentially in the past years and continues to develop as we speak

Lab Overview (Week 1):

- Review print, variables, and arithmetic operators
- Input statements
- Conditionals
- Functions
- Debugging
-