

## Quantum Computer Systems

### Contents

<b>1</b>	<b>Intro</b>	<b>2</b>
<b>2</b>	<b>Background</b>	<b>6</b>
2.1	Bits v. Qubits . . . . .	6
2.1.1	Boolean Circuits . . . . .	6
2.1.2	Qubits and Quantum Circuit Model . . . . .	7
2.1.3	QC: Architectural constraints . . . . .	8
2.2	Basic Principles of Quantum Computing (Ideal (noise-free) system) . . . . .	8
2.3	Noisy Quantum System . . . . .	11
2.3.1	Operator Sum Representation (OSR) . . . . .	11
2.3.2	Noises . . . . .	11
2.4	Qubit Technologies . . . . .	12
2.4.1	Architecture: Trapped-Ion Qubits . . . . .	12
2.4.2	Architecture: Superconducting Qubits . . . . .	14
<b>3</b>	<b>Quantum Application Designs: Computational Algorithms QC Can Solve</b>	<b>15</b>
3.1	General Features of Quantum Algorithms . . . . .	15
3.2	Query Model of Computation and Quantum Parallelism . . . . .	16
3.3	Quantum Computational Complexity and Cost . . . . .	16
<b>4</b>	<b>Quantum Systems: Overview</b>	<b>17</b>
<b>5</b>	<b>Quantum Programming Languages</b>	<b>17</b>
<b>6</b>	<b>Circuit Synthesis and Compilation</b>	<b>18</b>
<b>7</b>	<b>Noise Mitigation and Error Correction</b>	<b>18</b>

### References

- [1] Y. Ding, F. Chong. [Quantum Computer Systems: Research for Noisy Intermediate-Scale Quantum Computers](#). Springer Cham. Synthesis Lectures on Computer Architecture. 2020
- [2] [Crash Course: QM 1/2](#), [Crash Course: QM 2/2](#), [QC Explained](#), [QC w/QP](#), [How QC works \(Qubits\)](#)
- [3] [Wiki: QC](#), [Wiki: Quantum Logic Gates](#), [Wiki: Quantum Tunneling](#), [Quantum Superposition](#)

- [4] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-J. Wang, Simon Gustavsson, William D. Oliver. [Superconducting Qubits: Current State of Play](#). Annual Review of Condensed Matter Physics 2020 11:1, 369-395

## 1 Intro

- Simulating models of quantum mechanics (i.e. behaviors of elementary particles) is unfeasible on traditional computers because of the staggering number of all possible arrangements of electrons on small molecules.
- Advances in quantum chip manufacturing has given us small- and intermediate-scale quantum computer prototypes. This era where quantum computers are large and reliable enough to perform small computational tasks, i.e. the Noisy Intermediate-Scale Quantum (NISQ, John Preskill).
- The goal of quantum computers is to run real-world applications that classical computers have a hard time solving efficiently.

When we are going complexity analysis on a program or algorithm, has it always been in the context of classical computer models? How does complexity analysis change for quantum computing models?

- Quantum computing/**quantum algorithms**:
  - only computing model not bound by Church-Turing thesis on computability, i.e. a function on  $\mathbb{N}$  can be computed if and only if it is computable by a Turing machine. All computers can only be probabilistically faster than a Turing machine, but a quantum computer can solve computational tasks more efficiently than what was allowed in classical computation and complexity theory.
  - Computational problems implemented appropriately on a quantum computer can be solved exponentially faster than classical computers (Fourier sampling, factoring large integers, discrete logarithms, cryptography that relies on prime factorization (Shor's algorithm), unstructured database search).
- **Quantum Computer (QC)**:
  - A QC is a computing device that stores information in quantum bits (**qubits**), rather than traditional bits.
  - Qubits are transformed by exploiting properties found in [quantum mechanics](#). The behaviors of quantum mechanical systems, where particles are at very small energy and distant scales (i.e. atomic and subatomic), are explained by quantum mechanic theory.

I still don't understand how quantum mechanics theory translate into computational behavior in hardware? Is behavior now just based on formulas found in quantum mechanics? Is that not also computationally expensive in hardware and software?

- Traditional digital computers store information in sequences of 0 and 1 bits, and communication is handled by sending the sequence as electrical signals through integrated circuits. Our computers can handle billions of instructions per second because this process is deterministic and fast.

- Models of Quantum Computing (QC)

**Analog** : the state of a quantum system evolves gradually/*slowly* using quantum operations that *smoothly* changes the system. Resultant information encoded in the final system corresponds (with high probability) to the desired answer.

- The slow evolution of a quantum system allows it to remain at ground state energy. This approach is called an Adiabatic Quantum Computing.
- Quantum annealing is when these restrictions are lifted and a quantum system is allowed to interact with the thermal environment.
- Unclear if quantum annealing devices exist that can achieve universal quantum computation or speedups.

**Digital Gate-Based** : information is encoded onto a *discrete* and finite set of *qubits* and quantum operations are broken down into a sequence of a few basic *quantum logic gates*. The correct answer is given by digital measurement outcomes of the qubits.

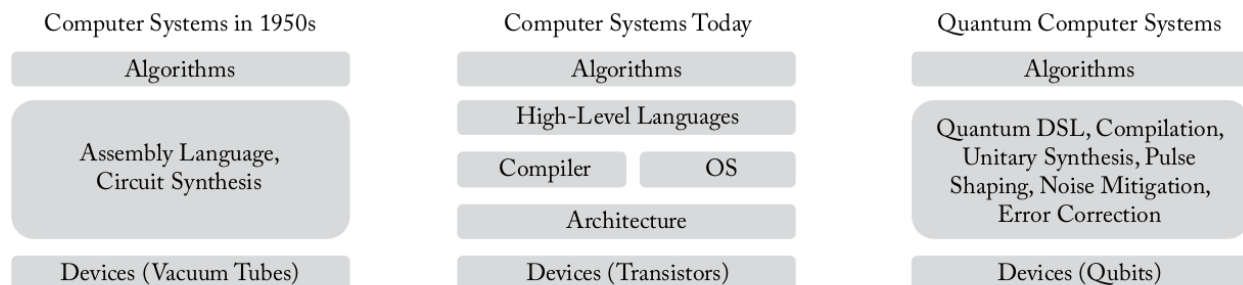
- Digital QC is more sensitive to noise than Analog QC; noise is non-negligible, and requires noise mitigating techniques to protect the information from qubit decoherence, imprecise control, and manufacturing defects.
- Discretization allows for discretization of errors and use of redundancy to encode information.
- NISQ digital quantum computers are QCs that deal with noise in this way.

**Measurement-Based** : a universal quantum computation model. Qubits are initialized in a cluster state, and computation involves measuring the qubits in the cluster state. Some measurements are possibly conditioned on the previous measurement outcomes of qubits. Each measurement accomplishes a quantum gate due to gate teleportation.

So is this just using quantum mechanic models in order to arrive at a solution quickly and with a high probability of being the “right” solution?

- Quantum Processor Unit (QPU) for Classical Computers

- QC hardware is referred to as QPU because it is envisioned as a hardware accelerator (or co-processor) for classical computers; similar in purpose to a GPU, a QPU’s purpose is to deal with quantum information.
- QPU does not fetch data or instructions on its own like a GPU.
- Architecture of Quantum Computer Systems is expected to rely on classical processing and classical control.



**Figure 1.2:** Architectural designs of classical vs. quantum computers. The abstraction layers for 1950s classical computing, today’s classical computing, and quantum computing are compared.

- In the NISQ era, a QCS cannot copy the layers of abstraction of conventional computer systems (middle); which is why they look like the designs of the 1950s computers, when resources were more strained. Quantum computers are physically large, expensive to make, limited in number of computing units, and demanding on power, just like how early computers were for their time period. NISQ era QCs have 50-1000 qubits, and optimizations are focused on every bit of the limited resources.
  - Conventional computers aim to suppress the noises caused by quantum mechanics in the transistor components (because they are so small); Quantum computers aim to harness the powers of quantum mechanics. The control apparatuses are fundamentally different between QCS and classical systems.
  - Tradeoffs and areas of optimization are reliably addressing and controlling qubits and correcting errors.
  - Classical computers are needed to control and assist quantum processors because control complexity becomes overwhelming as the number of qubits scale up; system-level automation can guarantee successful execution of quantum programs.
- **Quantum Technologies** include more than just quantum computation.
  - It is believed we will see similar scaling in quantum computers as to how 1960s Moore’s Law predicted the growth of digital computers. [Schoelkopf’s Law](#) predicts an exponential increase for superconducting qubits based on reported qubit coherence times (lifetimes).
  - **Qubits**
    - Every qubit doubles the computational space in which quantum machine operates.
    - Qubit resources are limited in today’s technology. The quantum machines today have high error rates and will likely for some time until qubit resources catch up. Other areas of study to prepare for such a future include:
      - \* The theory of quantum error correction codes will ideally (in the long term) support error-free quantum computations.

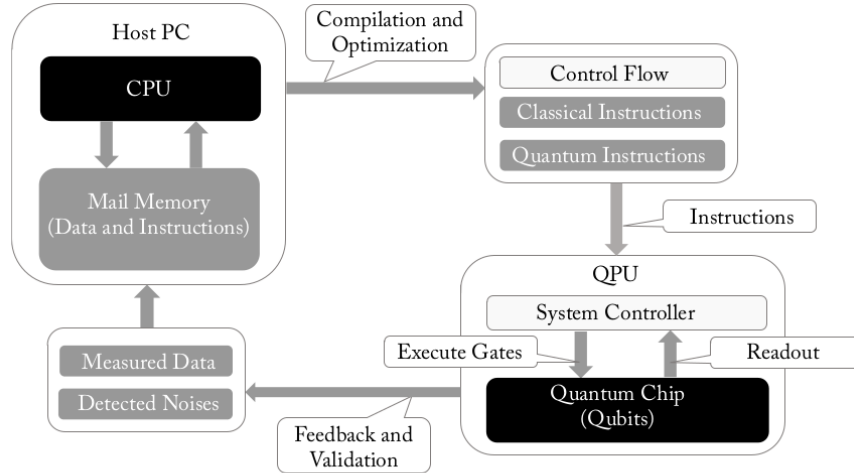


Figure 1.1: A QPU (quantum processor unit) and how it interacts with classical computers.

- \* error-tolerant algorithms, correctness of quantum algorithms.
- \* lightweight error-mitigating techniques.
- \* effects of noise on performance

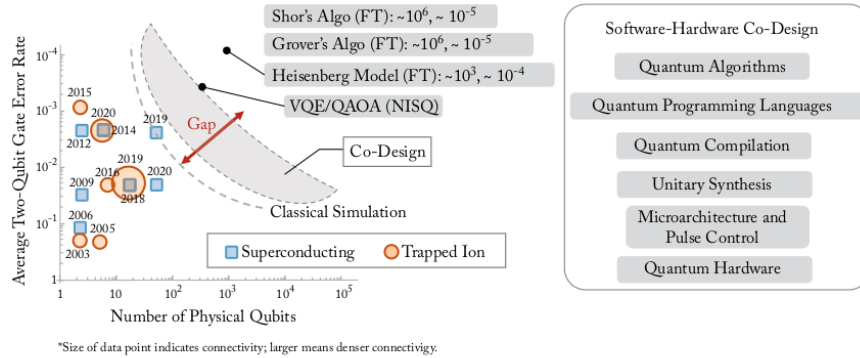


Figure 1.4: Status of qubit technologies [57–67]. Also drawn the gap between algorithms and realistic machines. Breaking abstractions via software-hardware co-design will be key in closing this gap for NISQ computers, hence the overarching theme of this book.

- Theoretical advances and quantum algorithms are at a point far ahead of where the current technology is. There is a gap between the machines we expect and the algorithms necessary to make full use of their power. Schor’s algorithm and Grover’s algorithm, for example, require machines much larger than what is currently practical; even smaller-scale quantum algorithms suffer from this gap.
- Software-architecture stack may close the gap with automated optimizations and co-design. The automation will also scale well when quantum programs scale up.

## 2 Background

### 2.1 Bits v. Qubits

#### 2.1.1 Boolean Circuits

- Boolean circuit Model of Computation:
  - information is stored and manipulated in bits
  - any computation can be done as a circuit of boolean logic gates.
  -
- von Neumann Architecture
  - cpu(alu/CONTROL UNIT),
  - ram
  - i/o DEVICES
  - eXTERNAL STORAGE
  - Modern CPUs are implemented in electric circuitry printed on integrate circuits, whose building block's are transistors.
  -
- Moore's Law: number of transistors per IC grew exponentially every few years since the 60s. But feature size is trending to a stop at a few nanometers (atomic level, where noises from quantum mechanical processes start changing the system).
- Reversible Computation
  - Quantum computers transform quantum bits reversibly
  - some bit operations are logically reversible, in that the output state uniquely determines what the input state was. ex. NOT gate. Or time-reversible: transformation done by a reversible circuit can be reverted by applying the inverse transformation, which always exists.
  - any Boolean circuit can be transformed into a reversible one, and then a quantum one by implementing a quantum Toffoli gate and replacing each bit with a quantum bit.
  -
- Randomized Computation
  - the physical world contains randomness, as commonly seen in quantum mechanics
  - randomness may be unnecessary and we can simulate randomized algorithms as efficiently with deterministic ones

- $n$ -bit probabilistic system (all  $p_b$  values must be non-negative and sum to 1):

$$\sum_{b \in \{0,1\}^n} p_b |b\rangle$$

The physical system is in one of these states. Once the random bits in the system are observed, then the state of the system is changed to reflect what was just learnt (laws of conditional probability).

### 2.1.2 Qubits and Quantum Circuit Model

- Quantum computing described as either randomized computation or reversible computation, each with extra features/conditions.
- Quantum Circuit Model
  - Quantum system  $\psi$  is an  $n$ -qubit system ( $\alpha_b$  is amplitude):

$$|\psi\rangle = \sum_{b \in \{0,1\}^n} \alpha_b |b\rangle$$

$\alpha_b$  can take any complex numbers, and their sum of squared values is 1. i.e.  $\sum_{b \in \{0,1\}^n} |\alpha_b|^2 = 1$

- *Superposition* : probability distribution across bit-strings. Example superposition state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

- *Entanglement* of qubits is the correlation between bits
- Note that random bits and quantum bits are fundamentally still different objects
- *Measure* (observe) the outcome of a qubit, measure all the qubits at the end of a circuit, just like with random bits. Bit-string  $|b\rangle$  is observed with probability  $|\alpha_b|^2$
- upon measurement, the state of the system collapses to the single classical definite value, and can no longer revert back to the superposition it was before.
- *Hadamard Transformation*: a single-qubit quantum gate: multiple results multiple results

$$H = \begin{cases} |0\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle, \\ |1\rangle \mapsto \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \end{cases}$$

Allowing Hadamard gates in a reversible circuit extends the circuit model over any functions allowed to be computed on qubits

- QM allows for a class of unitary transformations, not just Hadamard and Toffoli transformations.

- Amplitudes can interfere constructively (accumulate), or interfere destructively (cancel each other out)
- quantum circuit looks like reversible circuit, but the qubits are acted on by quantum gates (unitary transformations. )
- A qubit is a physical object integrated on the quantum chip (i.e. atom, superconducting gadget, etc.). A gate is pulse signals addressed to qubits (laser beam, microwave pulse, etc. )
- Classical arch: a gate is an electric circuite component on the CPU. A bit is a voltage signal along wires sent through gates and switches,

### 2.1.3 QC: Architectural constraints

1. Info in quantum state extracted through statistics from measurements. Outcomes are probabilistic, so quantum programs must ensure the desired bit-strings are observed with much higher probability compared to undesired ones.
2. No-cloning theorem: can't copy qubits; reads are done through measurements that likely alter data, and writes generally require complex state preparattion routines. Can't implement quantum analog of classical memory hierarchy, so current QC arch proposals generall have that transformations are applied directly to quantum memory.
3. entanglement / qubit-qubit interactions: incurs communication costs on quantum algorithms on NISQ machines; qubit-qubit interaction overhead is high.
4. NISQ machines highly sensitive to environment and control noise, and qubit decoherence (loss of quantum info) happens naturally from spontaneous loss of energy (loss of phase). These errors can accumulate if noot corrected or mitigated.

## 2.2 Basic Principles of Quantum Computing (Ideal (noise-free) system)

- Classical computing as operations under laws of boolean algebra. Quantum computation as operations under laws of linear algebra and probabilistic nature of quantum states.
- Behaviors of QC manipulated by quantum logic. QC described in in 4 postulates:
- **Quantum States postulate:** defines the superposition of each qubit.
  - A single-qubit quantum state (a superposition state),  $|\psi\rangle$ :

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

$$- \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1$$



- $\alpha, \beta$  are amplitudes of quantum state.
- superposition state can be rewritten as linear combination using basis states  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ :

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

- Computational basis states,  $|0\rangle$  and  $|1\rangle$ , are common. Other states

$$\begin{aligned} |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} & |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ |+\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) & |-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\ |i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) & |-i\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \end{aligned}$$

- **Composition Postulate of Quantum Systems:** generalizes quantum states to represent system of multiple qubits, and defines formally the entanglement property.

- In classical computation, an  $n$ -bit system has  $2^n$  possible states.
- In quantum computing, from superposition principle, the *joint state* of an  $n$ -qubit system should be a linear combination of  $2^n$  possible basis states. Example, 2-qubit system is linear combination of 4 basis states:  $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$
- Tensor product  $\otimes$  of smaller quantum system makes a larger quantum system:

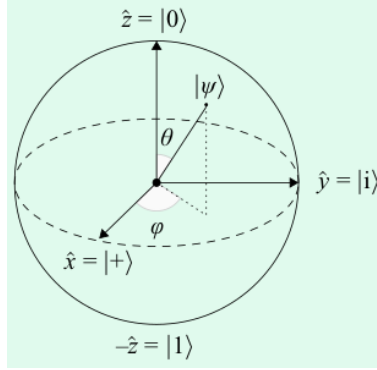
$$\begin{aligned} |\psi_0\rangle &= \sum_j \alpha_j |a_j\rangle & |\psi_1\rangle &= \sum_k \beta_k |b_k\rangle \\ |\psi\rangle &= |\psi_0\rangle \otimes |\psi_1\rangle \\ &= \sum_j \sum_k \alpha_j \beta_k (|a_j\rangle \otimes |b_k\rangle) \end{aligned}$$

(\*Can be written as:  $|a_j\rangle \otimes |b_k\rangle \sim |a_j b_k\rangle$ )

- Separable (product) states are those that can be written as tensor product of two quantum states.
- Entangled states are a class of multi-qubit states that can't be written as tensor product of two quantum states. Example, Bell state:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

- Qubit quantum state visualized in 3D is the Bloch Sphere:
  - \* 1-to-1 mapping from a pure single qubit state  $|\psi\rangle$  to a point on surface of Bloch sphere



- \* mixed-state is a probability distribution of several pure states, and all mixed states are inside Bloch sphere.
- \* No quantum states exist outside the bloch sphere.
- **Measurements postulate:** describes how much info can be read out from a quantum system, and effects of the measurement action to the system
  - The only way to read info out of a quantum system is by interacting with the system via measurement.
  - Measurement yields probabilistic outcome.
  - Measuring qubit  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , we observe basis state  $|0\rangle$  with probability  $|\alpha|^2$ , and basis state  $|1\rangle$  with probability  $|\beta|^2$
  - Measurement is irreversible and probabilistic. i.e. once measured,  $|\psi\rangle$  collapses into one of the two basis states, and the original quantum superposition is not recoverable.
  -
- **Quantum Gates postulate:** defines the logical operations that transform a quantum system.
  - unitary transformation is “norm-preserving linear transformation”
  - to ensure that both  $|\psi\rangle, |\phi\rangle$  are valid quantum states, we must have that  $|\alpha|^2 + |\beta|^2 = |\alpha'|^2 + |\beta'|^2 = 1$  when transforming  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  to  $|\phi\rangle = \alpha'|0\rangle + \beta'|1\rangle$
  - A valid logical transformation must map one quantum state to another quantum state.
  - Logical transformations are deterministic, and they are reversible, unlike measurement operators. No information is destroyed or leaked to the environment under unitary transformations.
  - Transformation doesn't account for effects of noise. Qubits Decohering to a classical state is incoherent and not reversible.
  - Single-qubit gate is a transformation of one point in bloch sphere to another, by rotating by an arbitrary angle along an axis.

- Multi-qubit gates have an entangling effect. Conditional gate: operation applied to one qubit is dependent on the state of the other qubit.

–

## 2.3 Noisy Quantum System

- Quantum systems must involve intentional (e.g. measurements) and unintentional (e.g. random perturbations) incoherence processes.
- probabilistic quantum state applies to noisy quantum system that likely produces a random distribution of quantum states.

### 2.3.1 Operator Sum Representation (OSR)

- any physical processes that can happen to a mixed quantum state can be written as a linear map (superoperator):  $\rho \mapsto \mathcal{E}(\rho)$
- unitary transformation (without interaction with the environment) can be written as an unitary operator:  $\mathcal{E}(\rho) = U\rho U^\dagger$
- $U$  is a unitary transformation.
- $U$  acting on both the system and environment (OSR):

$$U(|e\rangle \otimes |\psi\rangle) = \sum_k |e_k\rangle \otimes E_k|\psi\rangle$$

$$\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$$

- $E_k$  is the Kraus operators

### 2.3.2 Noises

#### 1. Amplitude Damping Noise

$$\mathcal{E}(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \quad E_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}$$

- quantum system losing energy (spontaneous emission of electromagnetic radiation for a atom can be modeled as amplitude damping, where  $\gamma$  is the probability of emission)
- as time goes by, a quantum state is exponentially more likely to undergo energy loss

## 2. Phase Damping Noise

$$\mathcal{E}(\rho) = E_0 \rho E_0^\dagger + E_1 \rho E_1^\dagger$$
$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix} \quad E_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}$$

- for some  $\lambda$  parameter between 0 and 1.
- Loss of quantum information without loss of energy.
- process where a qubit undergoes a phase flip (i.e.  $Z$  gate) with probability  $(1 - \sqrt{\lambda})/2$

## 3. Depolarizing Noise (stochastic Pauli noise): ?

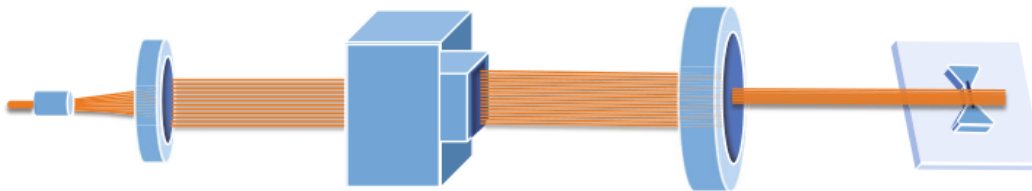
## 2.4 Qubit Technologies

- Classical equivalent to knowing how qubits work is knowing how transistors work.
- Leading qubit system technologies today (**trapped ion qubits, superconducting qubits, semiconductor spin qubits, linear optics, Majorana qubits, etc.**) have the potential for realizing scalable quantum computing. The book discusses just the first two.
- Qubit design follows Divincenzo Criteria
  1. scalable system with well organized qubits.
  2. ability to initialize qubits (prepare in computational basis)
  3. stability of qubits (long decoherence times)
  4. support for universal instruction set (single-qubit gates, CNOT, etc) for arbitrary computation
  5. ability to measure qubits (readout in computational basis)
- Tradeoffs and difficulty seen in quantum computer designs: initializing, performing gates, and measuring requires interactions between system and environment, but long decoherence times require isolating system from environment.
- Requires a physical system with potential to implement qubits and that demonstrates scalable architecture.

### 2.4.1 Architecture: Trapped-Ion Qubits

- make a qubit with an atomic ion (their internal energy levels exhibit great quantum mechanical properties). Ions are picked because they are charged particles that feel the forces exerted on them by the electromagnetic field.
- Two internal energy levels of an atomic ion (there may be more) represents the quantum states, one for  $|0\rangle$  and one for  $|1\rangle$

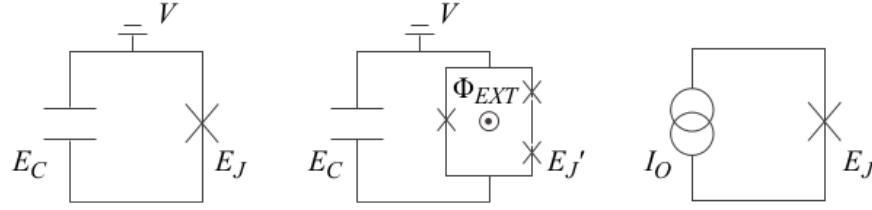
- *Optical Qubits*: the two chosen energy levels have a separation of  $10^{15}$  Hz (in the freq. of visible light), are from different orbitals (the  $s$  and  $d$  orbits) of the ion. Common optical qubits include:  $\text{Ca}^+$  (calcium),  $\text{Sr}^+$  (strontium),  $\text{Ba}^+$  (barium),  $\text{Yb}^+$  (ytterbium)
- *Hyperfine Qubits*: both energy levels are chosen from the  $s$  orbital of the ion. Energy separation is about  $10^{10}$  Hz (in the microwave spectrum). Common hyperfine qubits are:  $\text{Ca}^+$  (calcium),  $\text{Sr}^+$  (strontium),  $\text{Ba}^+$  (barium),  $\text{Yb}^+$  (ytterbium),  $\text{Be}^+$  (beryllium),  $\text{Mg}^+$  (magnesium),  $\text{Hg}^+$  (mercury),  $\text{Cd}^+$  (cadmium),  $\text{Zn}^+$  (zinc)
- The book references the  $^{171}\text{Yb}^+$  (70 protons, 101 neutrons) hyperfine qubit
- Measuring trapped-ion hyperfine qubit: state-dependent fluorescence, i.e. excite the qubits in the  $|1\rangle$  state  $s$  orbit into the  $p$  orbit, and measure the transition (decay, which emits photons, which is the fluorescence) back down to the  $s$  orbital with the  $|0\rangle$  state (lower energy level of the two)
- In an QC, a laser beam controls qubits. Lower state electrons are excited to higher states when the laser's frequency matches the separation between the two states (e.g.  $10^{10}$  Hz, state transitions controlled by microwave pulses for hyperfine qubits). Pro: low error rates, Con: difficult to focus on individual ions because microwaves are larger wavelengths. Alternatively, state transition via Raman transition, which will excite the electrons to the  $p$  orbital/state and decay it back to  $s$ ; con: spontaneous decay can be instantaneous.
- 2-qubit gate in a trapped-ion system is called the XX gate or the Ising gate. Entangling interactions achieved via [dipole-dipole](#) coupling between two ions.
- Ions are trapped in place and prepared to their initial states via [RF Paul trap \(quadrupole ion trap\)](#) which keeps the ions stationary.
- Trapped-ion QPU, such as the High Optical Access (HOA):



**Figure 2.10:** Schematics for a trapped ion QPU. After initialized from the optical source on the left, the laser are split into independently modulated beams, and then focused on the HOA trap on the right, providing individual controls over the array of ions in the trap.

### 2.4.2 Architecture: Superconducting Qubits

- “artificial atoms”
- implemented with macroscopic, lithographically printed circuit elements, which are parameterized and configured such that they exhibit atom-like energy spectra (“artificial atom”) with desired quantum mechanical properties.
- allows convenient design of qubits using existing integrated circuit (IC) technology
- *Josephson Junction*: electric circuit element in a superconducting qubit, that is an insulator between two superconductors. It is used to implement an anharmonic oscillator, which has unequal energy spacing. So for computation, we drive transitions between two (lowest) energy levels (represented as qubit states) without exciting other energy levels. Note that the anharmonicity value sets the limit on speed of gate pulses we can apply to a qubit.
- pairs of electrons in the superconducting material form bonds, *Cooper pairs*, which have spin of 0 (normal electrons have  $\pm\frac{1}{2}$  spin). The cooper pairs can tunnel through the insulator in quantized fashion (one at a time), which gives us the discrete energy levels needed for making a qubit.
- **Superconducting qubit state is related to the number of electrons Cooper pairs tunnelled across the junction.**
- Types of Superconducting Qubits:
  1. **Charge qubit**: computational qubit states defined by the number of Cooper pairs on a superconducting island. Examples: Cooper-pair box qubits, transmon qubits.
  2. **Flux qubit**: superconducting quantum interference device (SQUID) is used instead of a Josephson junction. the energy levels correspond to the integer number of superconducting flux quanta induced in the loop. The book references flux qubits.
  3. **Phase qubit**: computational states are defined using the quantum charge oscillation amplitudes across Josephson junction
- Measurement: Qubit readout done by dispersive readout, which determines qubit state by state-dependent frequency shift of a resonator coupled to each qubit. Changes in the frequency of the resonator can be used to probe qubit state without interacting with qubit.
- Single-Qubit Gates: Two kinds of controls: microwave control (control done by coupling the superconducting qubit to a microwave source (commonly referred to as charge drive or qubit drive) via a capacitor; single-qubit rotation along x and y axis) and flux control (single-qubit rotation along z-axis).



**Figure 2.11:** Types of superconducting qubits. **Left:** Circuit diagram for charge qubits (when  $E_J \leq E_C$ ) and transmon qubit (when  $E_J \gg E_C$ ), consisting of capacitor  $C$  and Josephson junction  $J$ . **Center:** Circuit diagram for a c-shunted flux qubit, where a junction is shunted with a number of junctions. **Right:** Circuit diagram for a phase qubit with current bias  $I_0$ .

- Two-Qubit Gates: interaction between two qubits is done by tuning the frequencies of the transmons so that the the energy exchange ahppens through capacitive coupling. Ex. resonance.

### 3 Quantum Application Designs: Computational Algorithms QC Can Solve

Topics:

- Quantum Information Procesing
- Quantum Parallelism
- Quantum Program Cost/Complexity
- Algorithms for NISQ

#### 3.1 General Features of Quantum Algorithms

- Quantum computing gives us the ability to encode exponential computational space into just a linear number of computational units, i.e. the state of an  $n$ -qubit quantum system is represented by  $2^n$  complex coefficients.
- Designing quantum algorithms means finding transformations on qubit states that yields a final measurement with a high probability of the desired outcome.
- Quantum Algorithms generally:
  1. efficiently encode info in small number of qubits.
  2. cleverly build up entanglement and interference during the algorithm
  3. design a final measurement that yields desired outcome with high probability.

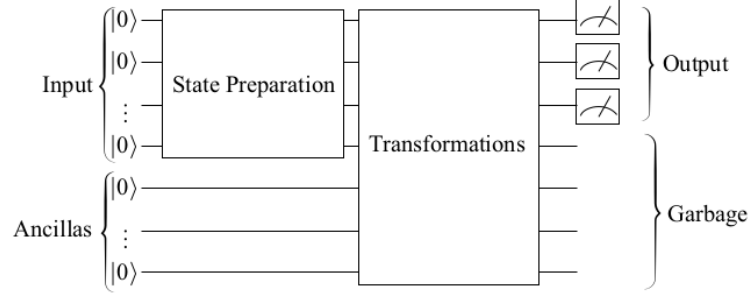


Figure 3.1: A typical quantum circuit implementation of a quantum algorithm.

### 3.2 Query Model of Computation and Quantum Parallelism

- Query algorithm is a measurement of how to efficiently learn properties of function  $f$  (black-box/oracle) of interest. They do so by passing the oracle inputs (query complexity) and analyzing the outputs.
- Classical query algorithms only accept a single input each time.
- Quantum query algorithms can pass several inputs as a superposition quantum state at one time, and then get a new superposition state as output.

### 3.3 Quantum Computational Complexity and Cost

- Cost is 1) device independent computational complexity, and 2) device-dependent implementation cost. Usually related, i.e. circuit complexity (number of gates) is related to running time (number of steps).
- Computational complexity of Quantum Algorithms usually analyzed in time and query complexity
  - Time complexity: time complexity of a unitary transformation  $U$  is related to the number of gates of the smallest circuit that implements  $U$ .
  - Query complexity: the number of times an algorithm needs to query a given black-box function (often called an oracle) to solve a problem
- The class of BQP (bounded-error quantum polynomial time) decision problems are solvable by polynomial-size uniform quantum circuit family. The relation of BQP to other computational classes is an open problem.
- Implementation Cost: Precision (quantum gates not necessarily accurate due to noise) and resource cost (qubit count, gate count, circuit depth, communication cost, space-time volume).
  1. qubit count (circuit width): number of qubits an algorithm needs and the dimensional limits in which the algorithm can operate.
  2. gate count: number of gates an algorithm needs. Estimate success rate of algorithm, as gate error is dominant noise in NISQ QCs.



3. Circuit depth: number of steps the algo uses, or rather, the deepest path in the circuit from input to output (e.g. “critical path”). This is related to running time of quantum circuit. Higher circuit depth correlates with lower success rate, because it means more gates (more gate noise), as well as higher likelihood of qubit decoherence.
4. Communication cost: due to physical limits, work must be done to “communicate” qubits (closer for interaction/entanglement) to enable actual computation.
5. spacetime volume: “space  $\times$  time” of algorithm, or rather the quantum volume for a quantum machine is no. qubits  $\times$  depth (also error rate and topology). Quantum volume can be used to compare performance to two machines.

See rest of Ch. 3 for Quantum Algorithms explained in depth.

## 4 Quantum Systems: Overview

- Hybrid approach to quantum-classical co-processing, seen in example theoretical quantum algorithms that still use classical operations with quantum subroutines.
- Quantum Compilers must consider: quantum instruction set, qubit communication (fully connected qubit devices are difficult to achieve, and even 2-qubit gates are error-prone), hardware noise (memory error from qubit decoherence, gate error from imprecise gate control), and available parallelism.

## 5 Quantum Programming Languages

- Concepts of data and operations for QC is different than those for classical.
  - Data (qubit states) is intrinsically probabilistic; Info stored in qubit can only be read out by measurement (irreversible), which yields a probabilistic outcome.
  - Operations on quantum data stored in one part of memory can affect data in another part of memory, due to entanglement.
  - Quantum data cannot be duplicated (no-cloning theorem)
  - Quantum states are fragile (sensitive to noise, operational error, and susceptible to decoherence)
- Quantum Algorithms typically involve a hybrid of classical and quantum processing
- Quantum PL design is approached is to adapt and augment conventional PLs (semantic and typing), to express new properties of quantum programs.
- Quantum assembly (QASM) is an early quantum language; it uses a linear sequence of gate instructions to describe programs. (low level, and being sequential makes it difficult to model complex classical control like looping). QASM extensions:

- Open source: IBM was the first to release a quantum cloud computing, and the QC community has grown substantially
  - OpenQASM is backended by IBM
  - ARTIQ was developed by the trapped ion community
- Most (HL) QPL are DSL, which lends itself well to building off (reusing) existing DSL designs. Its easier, but more general language that adheres closer to quantum information processing theory will help in development of new quantum algorithms.
- Functional QPL:
  - Quipper, Quaf, LIQuI $\rangle$ , Q#
- Imperative QPL:
  - Scaffold (embedded in C/C++)
  - ProjectQ
  - Quil (embedded in Python)
- Concern in QPL design: lifetime and scaling, because the NISQ systems are rapidly evolving and tools like PL need to keep up, because language, optimizations, compilation, and system design are intertwined. For the most part, quantum algorithms are verified with mathematical techniques because they are theoretical, so PL is more important for practical, efficient, and correct implementation, rather than new algorithm development.
- QPL design (and the supporting software toolchain) is still new, compared to hardware development and theoretical algorithm understanding.

## 6 Circuit Synthesis and Compilation

- Quantum Circuit Synthesis Techniques:
  - choice of universal ISA
  - single-, multi-qubit, qudit synthesis
  - exact and approximate synthesis

## 7 Noise Mitigation and Error Correction

- Noise Source Characterization
  - state and process tomography: reconstruct an unknown quantum state from the outcomes of a series of measurements. It is limited in noise estimating of a process due to SPAM (state preparation and measurement) error.

- random benchmarking: applying long sequences of gates and postponing measurement to the end in order to address gate error.
- Noise Mitigation can be done with qubit designs that prolong coherence/lifetime, more accurate gate ops, etc.:
  - Randomized compiling: inserting random gates in quantum circuit, converting coherent errors and non-markovian noise into stochastic Pauli errors, which are easier to correct for while preserving the logical operations of the quantum circuit.
  - Noise aware mapping: noise in qubits and in gates can vary by the hour. Compilers can take the daily variations into account and optimize using available data on this to increase probability of correct outcome.
  - cross talk aware scheduling through: connectivity reduction, qubit frequency tuning, coupler tuning.
- Error Correction (QEC): the goal is to achieve fault-tolerance through redundancy in qubits and turning quantum errors to classical errors.
-